Dictionary-based methods and their applications in biology and medicine

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Tatiana Lenskaia

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Daniel Boley, Adviser

May 2021

Acknowledgements

Foremost, I would like to express my sincere gratitude to my adviser, Prof. Daniel Boley for providing the unique opportunity to participate in the development of the novel interdisciplinary collaboration between computational researchers and biological scientists. I am very grateful for his advice and invaluable guidance during this research project. I want to extend my sincere gratitude to other members of my Preliminary Committee and my Final Committee: Prof. Chad Myers, Prof. Paul Jardine, Prof. Yuk Sham, Prof. Alan Love, and Prof. Louis Mansky. I am very grateful for their continuous support, encouragement, insightful comments, and hard questions. Modern challenges in biology and medicine cannot be address within one field or discipline. I greatly appreciate advice, experience, and expertise that all the Committee members generously shared during the research project. I learned a lot from this novel interdisciplinary collaboration, and I hope that this research helps advance it.

The work of the author was supported by the BICB fellowship, Interdisciplinary Doctoral Fellowship, and the Doctoral Dissertation Fellowship at the University of Minnesota. The author acknowledges the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this study.

I would like to express my special thanks to Prof. John Carlis, who is no longer with us, for generously sharing his experience and expertise in teaching, writing, and research and providing vital guidance in addressing complex challenges in science and life.

Dedication

To my family whose love, support, and resilience allow me to pursue a career in science

Abstract

This study proposes methods to explore genome organization and identify genome interactions that do not rely on annotations and aim to work on whole genome data. These methods use string matching between collections of dictionaries that depict genomes with different levels of resolution. Each dictionary represents a mapping of the complete genome data into a set of unique fixed-length segments. The methods are inspired by biological mechanisms including restriction-modification systems and CRISPR-Cas defenses that use exact matching. The use of this string-oriented approach might help researchers better understand biological mechanisms and avoid many of the drawbacks associated with annotations.

These methods shift the computational paradigm from looking for specific instances such as genes and other elements within a genome to "full-search" analysis without preconceived targets. We hypothesize that the development of efficient dictionary-based screening methods will lead to a better understanding of genome organization and genome interactions. The results of this study indicate that these methods can capture many biologically significant relationships not easily captured by traditional approaches. The results of this study contribute to (a) changing a computational paradigm for processing genome data; (b) developing new methods for analyzing genome organization and relationships between genomes; and, (c) identifying and evaluating potential genome interactions at a broader scale for biological and medical applications.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Genomes represent complex structures that are not designed by human. Text can serve as one of genome "simplified" representations that is convenient for storing and exploring genome sequences. Three-dimensional structure of molecules, methylation patterns, secondary structures, and chromatin states are examples of genome complexity that are usually hindered when genome is stored in a text format. We can understand the meaning of some fragments of genome texts e.g., genes and operons (i.e., clusters of genes that are controlled as a single unit) and even alter them. However, the entire organization of these texts is yet to be discovered. Researchers have done a large work in annotating elements that are known in genomes in attempt to link them to biological functions (Stein, 2001). Often these annotations are done in an automated way based on the observed similarity between genes in different organisms, and they might lack thorough biological validation (Prada and Boore, 2019).

During annotation process, alignment is often used to capture certain degree of similarity by analysing substitutions, insertions, and deletions. Most existing methods for sequence alignment are heuristics that do not guarantee to find the best possible alignment. However, these methods usually produce results suitable for the purpose of routine genome annotation. It is important to note that similarity analysis can result in spurious alignments that do not carry any biological meaning, especially

while comparing short sequences such as ancient DNA (de Filippo et al., 2018).

Moreover, structural similarity of genes does not always imply similar function. For example, the *UEV* genes are present in many eukaryotes including protozoa, animals, plants, and fungi. These genes are very conserved, and they perform activities associated with diverse biological roles including cell-cycle progression, cell protection, and participation in the elongation of polyubiquitin chains (Long, 2000). In addition, similar functions can be performed by structurally different genes. For example, genes that code for DNA ligases are very diverse in sequence (Williamson and Leiros, 2020). The function of DNA ligases is to join the phosphodiester backbone breaks of double-stranded DNA. Although annotation based on sequence similarity provides some guidance, it should be considered with caution. Annotations might be sparse, incomplete, inconsistent, and sometimes purely wrong (Koonin and Galperin, 2002). Also, annotation might create a false impression that unannotated regions of genomes are unimportant or even junk.

Gene-coding sequences can occupy from 80% of genome sequence in prokaryotes to about 2% in human genome (Taft et al., 2007). Previously, the remaining part of a genome was considered as evolutionary garbage or "junk DNA" (Ohno, 1972). However, genome-wide association studies (GWAS) (https://www.ebi.ac.uk/gwas/) demonstrated that many associations found with complex diseases such as diabetes, schizophrenia, and obesity are not associated with a particular gene (or a set of genes) and are often located in intergenic regions. The systematic exploration of intergenic regions in human genome and genomes of several model organisms (*Mus musculus*, *Drosophila melanogaster*, and *Caenorhabditis elegans*) by ENCODE project (Snyder et al., 2020) has indicated that these regions contain many important functional elements. Despite the controversy and complexity of elaborating a "practical" definition for functional elements (Guttinger and Love, 2020), ENCODE provided evidence of utility in genome regions that do not carry genes. In our opinion, the important

take-home message from performing this laborious project is the following: "not a gene" does not equal "useless." Also, not observing the measurable functional impact of a given genome element does not mean that it does not have a function at present, or have one in the past, or might have one in the future. Thus, genome regions with unknown function should not be labeled as junk and discarded from consideration due to our ignorance.

Unfortunately, we still know very little about the organization of genomes despite the large volume of annotated genomes available to date. It seems that describing and cataloguing genome parts might be useful for some applications, but it tells us very little about genome organization as a whole. We assume that a genome has holistic properties, and the arrangements and interconnections between parts make genome a lot more than merely the parts taken together. This property of emergence is present in biological systems (Mayr, 1982). Although some fragments of genome may not contribute to genome's purpose, e.g., endogenous retroviral sequences (Griffiths, 2001) may be considered as selfish in some sense. However, they might be signs of previous interactions that were critical for organism survival and adaptation. We would compare genome to an operating system that has a purpose in a holistic sense, even though it might have many elements with questionable immediate utility like computer viruses and obsolete computer programs that are no longer needed or are used only once in a while. We need to move a level up from the annotation level to a holistic view on genomes to see how all the parts fit and work together. The holistic view on genomes allows us better understand not only genome organization, but it also can help move a level up to explore genome-genome interactions at the integrative level. After careful consideration of all the benefits and drawbacks of the annotation approach, we decided to focus on methods that work at the sequence level and do not make any assumptions about the relative importance of genome components.

It is important to note that researchers have been able to harness biological mech-

anisms that work on the sequence level in genomes based on string recognition. One prime example are the CRISPR-Cas systems that constitute a defense mechanism in prokaryotes (Barrangou and Van Der Oost, 2013). These systems are now widely used for precise genome editing (Gaj et al., 2020). Researchers can re-engineer them for editing eukaryote genomes. Also, researchers have employed restriction enzymes (Arber, 1978) that are a part of restriction-modification systems in prokaryotes (Loenen et al., 2014) to cut genome sequences into smaller pieces, e.g., for convenience in sequencing.

Both these biological mechanisms work on genomes using string recognition via exact matching. It suggests that, despite a high mutation rate observed in microorganisms, exact matching appears to be suitable for conducting complex biological tasks such as adaptive immunity. Moreover, these biological mechanisms utilize the opportunity of parallel access to many parts of a genome in living cells. The biological mechanisms described above use different ranges of string lengths: 4-12 bp for restriction-modification systems (http://rebase.neb.com/) and 20-40 bp for CRISPR systems (https://crisprcas.i2bc.paris-saclay.fr/). For those mechanisms, genomes appear as a collection of strings of fixed length that match or do not match the string they search for.

Our work shows that viewing a genome as a collection of separate sets of strings of fixed length as string length varies is a convenient representation of a genome to explore the working principles of biological systems such as CRISPR-Cas systems. Importantly, these separate sets of strings are not independent. They are closely related since they are drawn from the same genome sequence. However, each set taken separately provides a different level of resolution for viewing the genome (a genome dictionary). This suggested to us that deeper understanding of the relationships between sets and basic rules they obey might be also useful to understand genome organization and interactions between genomes.

The main contributions of this study include the following:

(1) the development of a set theoretic approach and holistic dictionary-based methods agnostic of annotations for exploring complete genomes (Chapter 2);

(2) the development of theoretical and computational frameworks for modeling genome intersection (Chapter 3);

(3) the application of the methods developed here to study three types of intersection:

(a) Intersection within a genome and CRISPR-Cas autoimmunity (Lenskaia and Boley, 2020a) (Chapter 5);

(b) Intersection within a set of genomes to detect similarity between viruses (Lenskaia and Boley, 2020b) (Chapter 7);

(c) Intersection between two sets of genomes to analyze shared fragments between bacteria and viruses (Lenskaia and Boley, 2019, 2018) (Chapter 4 and Chapter 6).

This study explores principles of genome organization and genome interactions through the lens of the existing relationships between genome dictionaries. The view on genomes as a collection of dictionaries of strings of fixed length allows interpreting the existing relations between genomes in terms of relationships between these dictionaries. For each string length m, there is a complete dictionary of strings of length m that contains all possible strings of this length (the universal set or complete dictionary). Thus, each string is an element in the universal set. A genome accommodates a subset of strings from the universal set. Two genomes may or may not share strings, i.e., the subsets may or may not overlap. The amount of overlap between genomes that persists as m increases might be indicative of the existing biological relationships. For small string length m, the universal set is rather small, and genomes usually include all possible strings as embedded substrings. As m increases, only a small fraction of all possible strings can appear in a genome and an

even smaller fraction can appear in an intersection of two genomes.

We conducted a theoretical analysis of genome string sets using a set-theoretic approach. Hence, the results of this analysis were compared with the results reported in the literature (Chapter 2). The literature review indicated that empirical findings obtained in previous research corresponded to the theoretical conclusions. However, optimal string length for genome representation was usually found by trial and error, and the existing relationships between different levels of resolution were not taken into account. Also, previously there were no explanation and understanding why some string lengths worked better than others for particular applications. The developed theoretical and computational frameworks help close this gap.

The existing methods for string analysis are usually optimized for certain applications and do not provide enough flexibility for the exploration purposes. Accordingly, we developed a library of routines suitable for the exploratory analysis. We utilized randomly simulated genomes for testing and debugging the computational methods as we developed them. When we applied the methods to real genomes, we found that in some cases the relationships between real genomes were similar to the ones observed in simulated genomes, and in some cases, there were striking differences. Further exploration of these differences contributed to the development of the statistical modeling approach (Chapter 3). Also, a formula for estimating the optimal level of resolution for comparative genome analysis with respect to string length was developed.

We applied the theoretical and computational frameworks to addressing modern biological challenges including the investigation of functional properties of bacteria using their viruses as probes (Chapter 4), the study of autoimmunity potential in prokaryotes accumulated by CRISPR-Cas systems (Chapter 5), the exploration of host-parasite associations (Chapter 6), and the development of computational methods to predict epidemics (Chapter 7).

The accomplishment of these major goals contributes to (a) changing a computational paradigm for processing genome data from a partial to a holistic view; (b) developing novel computational, statistical, and mathematical methods for analyzing genome organization and relationships between genomes; and, (c) identifying and evaluating genome interactions using scalable computational methods for biological and medical applications.

Chapter 2

The foundations of dictionary-based methods

This chapter summarizes theoretical and computational results that lay the foundations of dictionary-based methods. The first section describes the developed theoretical framework to investigate string regularities in genomes. The second section summarizes results of the previous research. The third section characterizes the developed computational framework for the further exploratory analysis.

2.1 Theoretical framework

2.1.1 Notation and definitions

To simplify the exposition, the following notation is used:

a – size of an alphabet, $a = 4$ for DNA and RNA, $a = 20$ for proteins;

m – length of the sliding screening window for strings;

G – genome, $|G|$ equals size of genome G;

$D_m(G)$ – (genome dictionary of strings of length m) set of all distinct strings of

length m found within genome G;

U_m – (complete dictionary of strings of length m) set of all possible distinct strings of length m for the alphabet of size a, $|U_m| = a^m$.

Unique string or distinct string means a string that is different from all other strings in a given set over a given alphabet by at least one position in length or one letter in content.

$g_m^{max}(|G|)$ - (genome capacity for string of length m) maximum number of strings of length m that can be accommodated in a genome of size $|G|$. Genome capacity can be calculated using the following equation:

$$g_m^{max}(|G|) = \begin{cases} |G| - m + 1, & \text{if genome } G \text{ is linear.} \\ |G|, & \text{if genome } G \text{ is circular.} \end{cases} \tag{2.1}$$

We represent genome G as a sequence of dictionaries. Each dictionary contains all contiguous unique strings of length m obtained by a sliding window, and m varies in the range of string lengths that the genome can accommodate, i.e., from 1 to the genome size:

$\mathcal{G} = \{\ D_m(G) \mid m = 1 \ldots |G|\ \}$.

This study focuses the analysis of genomes. The same algorithms and similar analysis could be applied in principle to amino acid sequences and proteins, but this

is not addressed in this thesis.

2.1.2 Genome dictionaries

We represent a genome as a sequence of dictionaries of unique strings constructed for possible values of string length m. For a given genome G, we obtain its dictionary $D_m(G)$ by scanning the genome with a sliding window of length m, sequentially shifting it one nucleotide at a time. The results are assembled into a dictionary of unique strings ("keys") together with an optional count ("values") of how many times that string appears in genome G (Figure 2.1). Depending on the desired end computation, the values stored in the hash table can be counts (how many times the given string appears in the genome) or positions (start positions of the string in the genome), or else can remain empty if we simply wish to store the presence or absence of the strings. If a genome consists of several replicons then the dictionary can include information about counts in each replicon separately. In this case, it is possible to use a list of counts or a more sophisticated data structure implemented in Python as a dictionary of dictionaries to facilitate the computations.

Figure 2.2 shows the size of $D_m(G)$ as m varies for a toy circular and linear sequence of length 100 bp. The size of $D_m(G)$, a genome dictionary for a given window size m, is bounded above by (a) the number of all possible unique strings of a given length m (i.e, $|U_m| = 4^m$) and (b) genome string capacity for a given string length m, i.e., $g_m^{max}(|G|) \sim |G|$.

The number of possible unique strings of length m in DNA/RNA alphabet grows exponentially $|U_m| = 4^m$ as m increases. Table 2.1 compares the size of the complete dictionary U_m for various values of m to the size of genomes for different organisms. For $m > 19$, most genomes are not capable of accommodating all possible unique strings.

There is a turning point m^*: $m^* \approx log_4 g_m^{max}(|G|)$. This turning point defines the

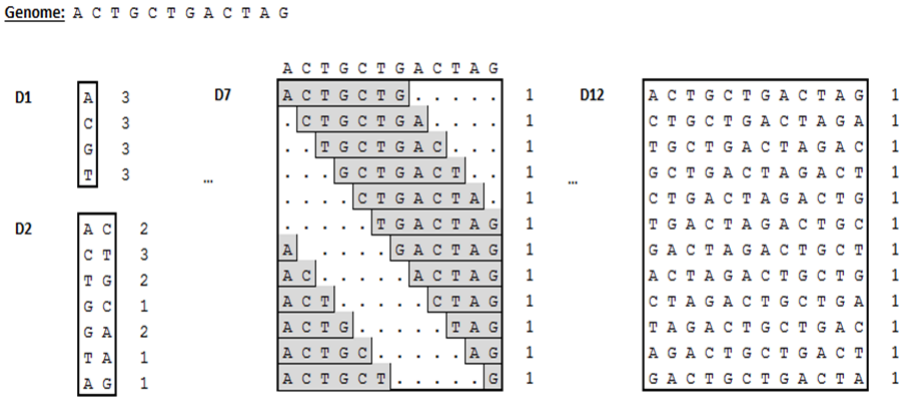

Figure 2.1: For a given "example" genome (assuming the genome is circular), we illustrate sequentially constructing a set of dictionaries of unique strings for lengths 1, 2, 7, 12 (the maximal window possible equal to the size of the whole genome). The black frames contain dictionary keys (unique strings found in the genome); the accumulated frequencies of the unique strings in the genome are shown next to each frame.

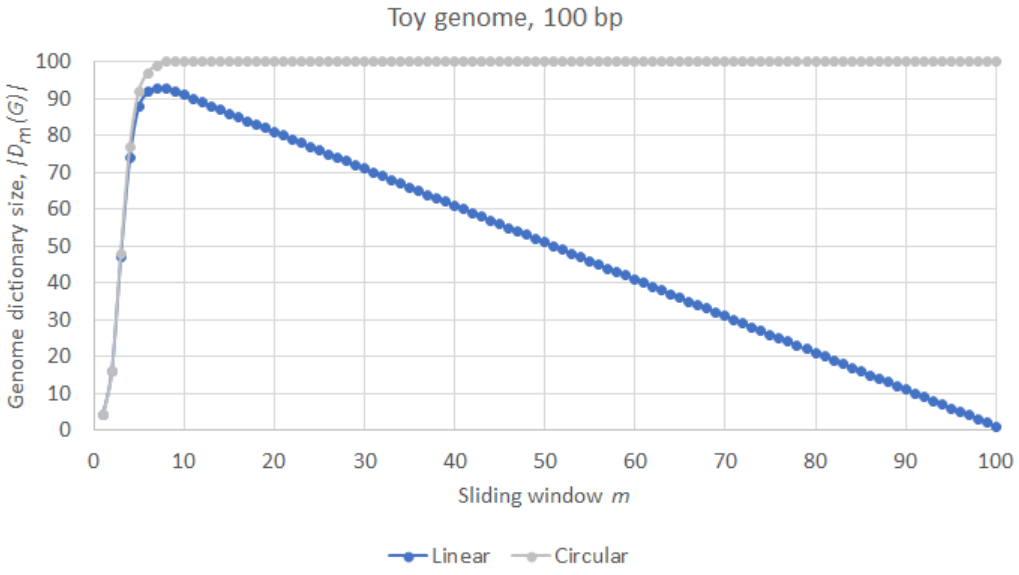

Figure 2.2: The relation between string length m and genome dictionary size for a toy circular and linear genomes.

Table 2.1: Comparison between the complete dictionary size and genome size for different organisms.

Screening window m	$\lvert U_m \rvert = 4^m$	Genome size, Mbp	Organisms
6	4096	0.0041	viruses
7	16384	0.0164	viruses
8	65536	0.0655	viruses
9	262144	0.2621	prokaryotes
10	1048576	1.0486	prokaryotes
11	4194304	4.1943	prokaryotes
12	16777216	16.7772	prokaryotes
13	67108864	67.1089	eukaryotes
14	268435456	268.4355	eukaryotes
15	1073741824	1,073.7418	eukaryotes
16	4294967296	4,294.9673	eukaryotes
17	17179869184	17,179.8692	eukaryotes
18	68719476736	68,719.4767	eukaryotes
19	2.74878E+11	274,877.9069	eukaryotes

relationships between the limiting factors. For string length m that is less than m^*, the size of the genome dictionary is primarily bounded by the size of the complete dictionary since the genome can accommodate much larger number of strings than the total number of distinct strings possible for a given length m, i.e., $|D_m(G)| \leq |U_m| << |G|$. For example, for $m = 2$, $|U_2| = 4^2 = 16$ while genome size of even a small virus is at least several thousand of nucleotides. Thus, the corresponding genome dictionary is likely to contain all possible unique strings of length 2 and has size that is equal to the size of U_2. For string length m that is greater than m^*, the main limiting factor is genome size since the size of the complete dictionary is much larger than genome size (see Table 2.1), i.e., $|D_m(G)| \leq |G| << |U_m|$. For example, for $m = 100$, $|U_{100}| = 4^{100} > 10^{60}$ and the size of the complete dictionary exceeds the size of genome for most organisms (see Table 2.1). In addition, the presence of a number of repeats in genomes affects the size of genome dictionaries. Thus, the observed genome dictionary size is often less than the number of strings that can be potentially accommodated in a given genome (i.e., the genome string capacity).

The computational complexity for constructing a dictionary for genome G is linear in time and space with respect to the size of the genome G (see Algorithm 1). Our method is unsupervised in that it does not depend on any annotation or other metadata. The dictionary is implemented using a hash table. We extract the string of length m at each position within a genome using a sliding window and then calculate its corresponding hash value. It is possible to use a "rolling hash" to compute all these hash values (after the first one) in time independent of m (Karp, 1987) even though this optimization was not needed for the experiments reported here. The observed time complexity was found to be dominated by the size of genome. However, this optimization might be essential for long strings and large eukaryote genomes.

Algorithm 1 Compute $D_m(G)$

Require: $0 < m \leq |G|$

$D_m(G)$ is an empty dictionary (i.e., it does not contain any key-value pairs)

if G is circular
    concatenate G with the first $(m-1)$ nucleotides of G
end if
$lastpos = |G| - m + 1$

for each position i in G starting from the first one to $lastpos$
    extract string of length m starting in position i
    if the extracted string is present in $D_m(G)$ as a key
        increment the corresponding value by one
    else
        add the extracted string to $D_m(G)$ with the corresponding value of 1
    end if
end for

return $D_m(G)$

2.1.3 Genome intersection

To find strings of a given length m shared between two genomes, we need to compute an intersection between two sets of dictionary keys by finding entries present in both dictionaries. We scan all the unique strings in the smaller genome dictionary, marking those that also appear in the larger genome dictionary. Using hash-tables, the computational complexity of this process is linear in relation to the size of the smaller dictionary.

When the string length m selected for constructing dictionaries is small enough, the keys in the two dictionaries are most likely to be identical, and they likely include all possible unique strings present in U_m (Figure 2.3). As we increase the string length m, the size of the complete dictionary U_m grows exponentially and the variety of strings present in a given genome becomes primarily limited by the length of the corresponding genome. Therefore, it becomes more likely that two different genomes share only a small number of strings from U_m or even share no strings.

To simplify the exposition, we use the following notation:

G_i – i-th genome, $|G_i|$ equals size of i-th genome, where $i = 1, 2$

$D_m(G_i)$ – set of all distinct strings of length m found within genome G_i (a genome dictionary)

$I_m(G_1 \cap G_2)$ - intersection between genome G_1 and G_2 that contains distinct common strings of length m in these genomes:

$$I_m(G_1 \cap G_2) = D_m(G_1) \cap D_m(G_2) \tag{2.2}$$

The size of the intersection $|I_m(G_1 \cap G_2)|$ is equal to the number of distinct strings shared between the two genomes.

To compare the proportion of the parasite genome integration into host genomes

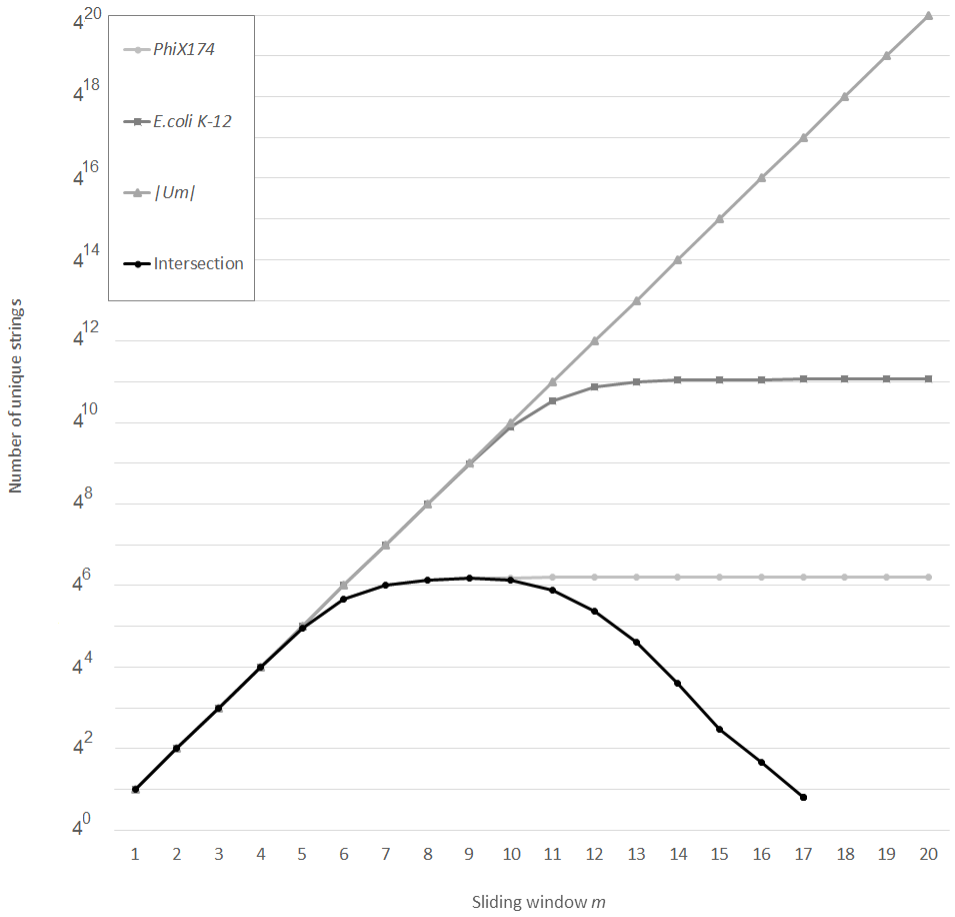

Figure 2.3: Size of the complete dictionary, the dictionaries constructed for two organisms, *E.coli K-12 MG1655* (NC_000913) and *Coliphage phi-X174* (NC_001422), and their intersection as m varies.

for different parasites, we compute the intersection ratio. This ratio can take values from 0 to 1.

$I_m^{P_j}(H_i)$ - intersection ratio of parasite genome P_j within host genome H_i:

$$I_m^{P_j}(H_i) = \frac{|D_m(H_i) \cap D_m(P_j)|}{|D_m(P_j)|} \tag{2.3}$$

Intersection ratio is equal to the number of distinct strings shared between the host and parasite genome dictionaries relative to the number of distinct strings present in the parasite dictionary.

2.1.4 Intersection matrix

We use matrix representation to organize and visualize the results of large-scale screening research and store information about the intersection between genome dictionaries. This representation allows us to use matrix operations to quickly access and summarize captured information about genome interactions for various purposes including visualization, clustering, and information retrieval. We use multidimensional arrays to accommodate information obtained from several screening resolution levels.

For example, given a collection of prokaryotes and viruses, we use the method described in the previous section to compute the intersection ratio between each prokaryote and each virus (see Chapter 4 and Chapter 6). The result is a rectangular matrix whose i, j-th entry is the intersection ratio computed between i-th prokaryote and j-th virus using equation (2.3).

Alternatively, the intersection matrix can store the intersection string counts for a collection of similar organisms, e.g., viruses. Each cell of this square matrix is the number of strings shared between the corresponding dictionaries of the i-th and j-th

virus in this collection (see Chapter 7).

Many entries in these matrices will be zero, hence we can use an efficient sparse matrix data structure for efficient storage and convenient matrix manipulations. This representation allows us to use linear algebra methods to analyze intersection and to capture and visualize important information (see Chapter 7).

2.2 Previous research

We seek methods that allow researchers to explore genome as a whole. In this respect, k-mers represent a promising approach that is capable of analyzing big genome data (Manekar and Sathe, 2018). Previousely, the development of shotgun sequencing technologies required effective methods for genome assembling from reads. Fast identification of overlapping short reads using de Bruijn graphs (Compeau et al., 2011) made possible to streamline genome assembling. K-mers are used not only for genome assembling but also for other tasks including sequencing quality check, identification of repeats, alignment seeds, and metagenome comparison (Manekar and Sathe, 2018). For example, the coverage of sequencing (how many times each fragment was sequenced) contains important information that allows researchers to reduce sequencing errors by filtering out k-mers that appear with frequencies that are too shallow in comparison with the coverage depth.

For many k-mers approaches, comparison is based on the distance between vectors of frequencies (Vinga and Almeida, 2003). To have vectors of frequencies, it is necessary that genomes shares those strings. For short strings, one can store frequency counts in genomes for all possible unique strings, but, as the string length increases, only frequencies for shared k-mers can be compared between genomes.

However, the question of suitable k-mer length persists. In all these applications of substring counting, the main question is what length is appropriate for a given

task. There is no one length that fits all. In many cases, suitable length has been detected by trial and error. Unfortunately, these empirical findings are relevant only for the particular data. The empirical process needs to be repeated for each new dataset, and it is poorly generalizible.

Moreover, the existence of biological mechanisms that work on different levels of resolution in genomes, e.g., CRISPR-Cas systems and restriction-modification systems, indicates that all possible sets of strings that a genome can accommodate need to be taken into account. We assume that understanding the relationships between these string sets can help us better understand genome organization and genome interactions.

The existing string methods for analyzing genome interactions, e.g., the longest common substring (Hirschberg, 1977), alignment (Altschul et al., 1997), and dot-matrix display (Gibbs and McIntyre, 1970), have some shortcomings. The initial dot-matrix method had a high cost $O(|G_1| \cdot |G_2|)$, where G_1 and G_2 denotes two genomes that are analyzed. The computational complexity of dictionary-based methods is $O(|G_1| + |G_2|)$. We have improved the performance of dot-matrix display by computing the intersection between genomes before plotting the matches (see Chapter 5, Figure 6.5). In this case, the cost of the dot matrix display is $O(|G1| + |G2|+$ [number of matches found]). When window size is too small, there are too many matches and the new dot matrix display is also expensive, and the matrix display is basically one big full matrix. But if window size m is more than 10-15 bp, there are fewer matches between genomes, and the dot matrix display becomes very fast since the matrix is sparse.

Both alignment and the longest common substring methods are limited to finding the single best overall match. The longest exact match often reflects the most recent genetic exchange event. Point mutations and insertions/deletions can obscure long shared fragment from being captured by the longest exact match method. Finding all

matches of a given length can preserve far more of the possible signs of the interaction between organisms.

2.3 Computational framework

Many existing methods are biased towards fulfilling specific search queries, which started with early examples such as the search for motifs (Schuler et al., 1991) and homologs (Altschul et al., 1997). It is typical to make software scalable by specializing it for optimal performance on certain narrowed tasks. Current computational paradigms often sacrifice "exhaustive-search" capabilities by utilizing heuristic algorithms such as BLAST (Altschul et al., 1997) and setting up arbitrary filters for the elements in question.

The usual method to optimize software tools is to disregard everything that is not directly related to what the tools are searching for. Although this method of optimization has immediate benefits for serving the target queries, there are key drawbacks to these methods. Optimization through discarding is possible only if we know what we are looking for. If this is not the case, an exhaustive search is preferable. Therefore, any "full-search" approach must only use "unbiased" ways of optimization that consolidate and preserve the underlying exhaustive-search capacity of the initial algorithm. Many researchers might have a biased perspective on the genome because of the narrow focus on specific genes and other elements; the ability of methods to look at the whole genome is important for unbiased analysis.

We seek alternative approaches to overcome the shortcomings of current methods in analyzing genome data. One of our primary goals is to define sensitivity and specificity trade-offs to capture signals of biologically important genome interactions in unannotated complete genomes. This has led us to consider basic string methods for "bare" genomes. Strings are universal elements and essential building blocks of

genomes. One can factorize genomes into these blocks and restore genomes by putting blocks back together to get an original object without loss. Moreover, aggregated information from all strings within a genome provides vast opportunities for analysis in a "full-search" paradigm.

String methods are powerful tools for initial genome assembly (Pevzner et al., 2001) and further analysis (Sievers et al., 2017). In particular, strings can help identify genome interactions. Long string matches (a) might be a distinctive sign of biological interactions when the probability of random coincidence is negligible, (b) can be done efficiently and hence be scaled to large amounts of data, and (c) can be done unsupervised to identify potential, previously unknown, interactions. Constructing a dictionary of strings is a convenient method of genome representation (Vinga and Almeida, 2003). In the previous research dictionaries were used to analyze individual genomes (Castellini et al., 2012).

In this study, we have developed a set-theoretic approach and created dictionary-based methods for analyzing the relationships between genomes using the intersection between dictionaries. We have found several applications of the dictionary-based methods in identifying biologically important interactions between genomes and made feasible computations for biological and medical applications. The proposed dictionary-based approach is very powerful, and it integrates the advantages of various string methods including k-mers, unique strings, and the longest common substring. We hypothesize that efficient dictionary-based screening methods for identifying genome intersections will provide important insights into genome organization, relationships between genome composition and phenotypes, and interactions between genomes.

The existing k-mer methods can be divided into the following broad categories that use: hash tables, e.g., Jellyfish (Marçais and Kingsford, 2011), suffix arrays, e.g., Tallymer (Kurtz et al., 2008), and sorting, e.g., KMC3 (Kokot et al., 2017). Often

the implemented optimization steps do not allow enough flexibility for the purpose of exploratory analysis. For example, integration of additional parameters that need to be stored for each k-mer can be problematic. Also, necessary adjustments of data structure are not always possible. For example, metadictionary or consensus dictionary can be implemented as a dictionary of dictionaries.

For the purposes of our exploratory analysis, we had to create a library of routines for unsupervised identification of biologically important genome interactions using the developed dictionary-based approach. Each routine is implemented as a separate module with maximum universality and compatibility like a "Lego" brick in order to facilitate fast and convenient assembly of necessary blocks to validate a hypothesis or solve a specific biological task. Each module is simple, easily modifiable, vastly connectable, and minimally specialized so they can solve a task but not be restrictive in terms of universality. They will preserve "exhaustive-search" capabilities and work reasonably fast considering that flexibility is preferred over optimality.

This library is intended to create a prototype software and a concept model/pipeline that is sufficient to validate a hypothesis and serve as a proof of concept. A model made of "conceptually uniform" modules is easy to create, operate, and modify. It provides flexibility that often is lacking when attaching and connecting "of-the-shelf" tools with many embedded optimization features. If the hypothesis is valid and the concept is viable, this prototype software or model will serve as a blueprint and be optimized and implemented as a solid "industrial" pipeline.

The project also has capacity for integrating state-of-the-art software solutions if they are suitable to substitute a specific module. The software will be flexible to satisfy possible requirements, specialized enough to solve a certain task reasonably fast, and universally connectable for a vast variety of future applications.

We utilized simulated genomes to test and debug the developed modules. This allowed us to vary sequence size and content without downloading large amounts of

data from online databases and storing them locally. Simulated sequences could be generated on the go in the necessary quantity and discarded when no longer needed. Also, simulated genomes allowed us to avoid dealing with unknown or partially recognized nucleotides that are often appear in genomes of organisms. In addition, the interpretation of the results for real organisms in comparison to the simulated genomes can be much more difficult.

Then, we applied our methods to bacterial and phage genomes as model systems since genome interactions are very common between bacteria and their viruses. Identifying the genome intersection between bacteria and viruses is seminal for exploring the functional and structural organization of these organisms (Brüssow et al., 2004; Touchon et al., 2016; Lenskaia and Boley, 2018). In the future, our methods can be adapted to exploring large eukaryote genomes.

Chapter 3

Statistical modeling

We can expose candidate evolutionary relationships between the genomes through pure $in-silico$ processing by comparing the size of the intersection of dictionaries of two different genomes with the intersection size that we would expect from random unrelated genomes. We develop a statistical modeling approach to determine the range of appropriate window sizes for a dictionary comparison. An appropriate window size must be long enough to avoid string overlaps by pure random chance (specificity), but short enough to capture relations between organisms (sensitivity). We demonstrate the application of statistical modeling to determine the appropriate window size to capture the genome interactions in exemplar bacterial and phage genomes.

We first determine the null-hypothesis to evaluate the probability of obtaining a non-empty intersection between two randomly generated genomes. Since the strings of length m come from a sliding window, they are not statistically independent, so they cannot be modelled by a simple statistical distribution such as a multinomial distribution. Thus, we use a numerical simulation.

3.1 Theoretical estimate

Our goal was to develop a theoretical model that would describe the probability of spurious overlaps between genomes $G1$ and $G2$ as the string length m varies. We denote $n_i = |D_m(G_i)|$ and $p_i^A, p_i^C, p_i^G, p_i^T$ probabilities of observing nucleotides A,C,G, and T in any individual position under IID (independent and identically distributed) model in genome G_i respectively, where $i = 1, 2$.

Therefore, the probability p of a single nucleotide match assuming that nucleotides are independent is the following:

$$p = p_1^A \cdot p_2^A + p_1^C \cdot p_2^C + p_1^G \cdot p_2^G + p_1^T \cdot p_2^T \tag{3.1}$$

And conversely, the probability of a single nucleotide mismatch is $1 - p$. For example, if the distribution of nucleotides is uniform then $p = 0.25$.

The probability of a match between two strings of length m equals p^m. Thus, the probability that two strings of length m do not match is $(1 - p^m)$. Moreover, the probability of no match between a particular string of length m and a set of n_1 strings of length m is equal to $(1 - p^m)^{n_1}$. Thus, the probability that no string out of a set of n_2 strings, each of length m, matches any string out of a separate set of n_1 strings, each of length m:

$$P[D_m(G_1), D_m(G_2)] = [(1 - p^m)^{n_1}]^{n_2} = (1 - p^m)^{n_1 \cdot n_2} \tag{3.2}$$

To compute this probability, we use the following formula:

$$P[D_m(G_1), D_m(G_2)] = e^{n_1 \cdot n_2 \cdot \ln(1 - p^m)} \tag{3.3}$$

In case of a skewed distribution of nucleotides, it is possible to use GC-content to estimate the probabilities of observing the nucleotides in genomes. We assume that the numbers of nucleotides in a genome satisfy the the following two conditions: (1) the numbers of four nucleotides add up to the genome size, i.e., genomes do not contain any symbols other than four nucleotides A,C,G,T (there are no unknown or partially recognized nucleotides), and (2) the number of Gs approximately equals the number of Cs, and the number of As approximately equals the number of Ts.

Thus, we denote GC-content in genome G_i as GC_i, and we can estimate probabilities of observing the nucleotides in genome G_i as follows: $p_i^C = p_i^G = \frac{GC_i}{2}, p_1^A = p_1^T = \frac{1-GC_i}{2}$, $i = 1, 2$. Then, we use these probabilities to calculate the probability of a single match using equation (3.1):

$$p = \frac{1 - GC_1 - GC_2 + 2 \cdot GC_1 \cdot GC_2}{2} \tag{3.4}$$

Figure 3.1 depicts the relationships between GC-content in two genomes and the probability of a single match. The graph is a hyperbolic paraboloid (see equation (3.4)), and it reaches a saddle point when GC-content equals 50% in both genomes with the corresponding value for the probability of a single match equaling 0.25.

Also, Figure 3.1 demonstrates that the probability of a single match reaches its maximum of 0.5 when both GC-content are extremely skewed in the same direction, i.e., either both are 0% (both genomes contain only As and Ts) or both are 100% (both genomes contain only Gs and Cs). These extreme cases are not observed in genomes of organisms since genomes contain all four nucleotides. Therefore, even in very skewed genomes of organisms, the probability of a single match is always less than 0.5. Thus, we can use the maximum value to get the lower bound for the probability of no match regardless of the observed skewness of GC-content in the

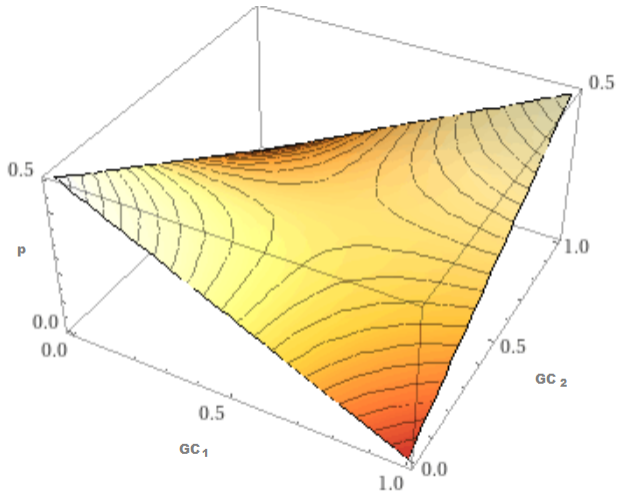

Figure 3.1: The relationships between GC-content in genomes and the probability of a single match.

genomes. This allows us to eliminate GC-content from consideration and adjust the formula:

$$P[D_m(G_1), D_m(G_2)] > (1 - 0.5^m)^{n_1 \cdot n_2} \tag{3.5}$$

3.1.1 Simulated genomes

We utilized simulated genomes to test and debug the developed methods. Also, we applied the simulated genomes to validate the statistical expectation of having common strings between genome sequences for different string lengths. *Simulated genome* is a randomly generated sequence of nucleotides of the specified size. This allowed us to vary sequence size and content without downloading large amounts of data from online databases and storing them locally. Simulated sequences could be generated on the go in the necessary quantity and discarded when no longer needed. Also,

simulated genomes allowed us to avoid presence of unknown or partially recognized nucleotides that are often appear in sequenced genomes of organisms. In addition, the interpretation of the results for simulated genomes is easier than for genomes of organisms since the simulated genomes share fragments only due to spurious overlaps that do not imply any evolutionary and biological relationships.

3.2 Empirical observations

To obtain estimates of the occurrence of non-empty intersections by chance for a given pair of simulated genomes, we repeat the simulation of the intersection of genomic dictionaries 10,000 times from randomly generated synthetic genomes of the same size and GC-content. This process is carried out for each length m of interest. Figure 3.2 compares the sizes of the intersection over a range of string lengths m (1 to 40 bp) for *Escherichia coli O157:H7* and two different phages. Both phages are known to infect this bacterium (Cowley et al., 2015). Figure 3.2 shows the intersection between the pair of real organisms compared to the results of 10,000 simulation runs. For the simulations, the maximum and minimum intersection found for each string length plus one sequence of intersections for one typical case are shown. Figure 3.2A shows an example of a phage with a statistically significant overlap at longer string lengths. For these longer string lengths, finding any intersection between two unrelated genomes would be extremely unlikely ($p < 0.0001$). Figure 3.2B shows an example of a phage whose overlap with *E.coli* is statistically indistinguishable from random.

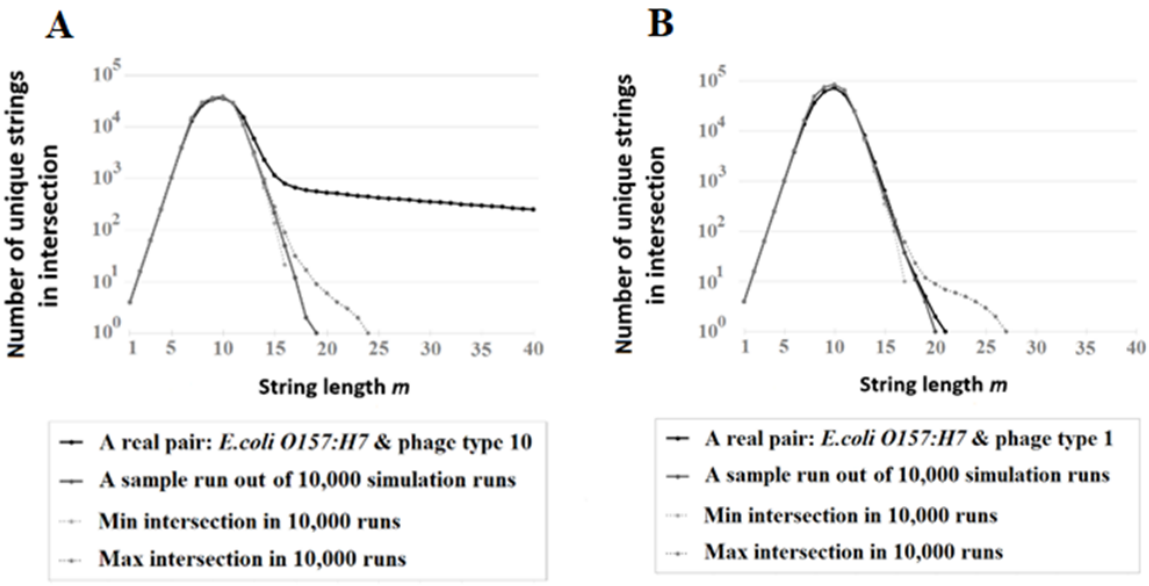

Figure 3.2: Each curve represents a number of strings in intersection between dictionaries of real genomes and their simulations in a string length range from 1 to 40 bp constructed for *Escherichia coli O157:H7* (5,498,450 bp) and one of the two its phages: (A) typing phage 10 (39,234 bp); (B) typing phage 1 (88,531 bp).

3.3 Simulation experiments

Biological interactions between bacteria and viruses that involve sharing genetic material could manifest themselves as long common strings between their genomes. Computational methods for detecting interactions must work on noisy biological data. Screening computational methods based on exact matching avoid ambiguity and do not depend on any annotation or metadata, and they can be implemented efficiently *in silico*. To estimate sensitivity and specificity of the screening window, we explore several factors (GC%, phage and bacterial genome size) that might influence the threshold for statistically significant intersections of the screening methods. We demonstrate simulation results for uniform distribution and possible extreme skewed cases for GC-content combinations and different genome sizes:

(1) phage – 12.5%, bacterium – 25% (Figure 3.3A);

(2) phage – 50%, bacterium – 50% (Figure 3.3B);

(3) phage – 12.5%, bacterium – 75% (Figure 3.3C).

Each probability was estimated based on 1024 simulation runs. On each run, a pair of randomly simulated genomes (bacterial and viral) with the given values of size and GC-content were analyzed. The number of distinct strings in their overlap was computed for a given screening window size. The empty intersection means no overlap between genomes.

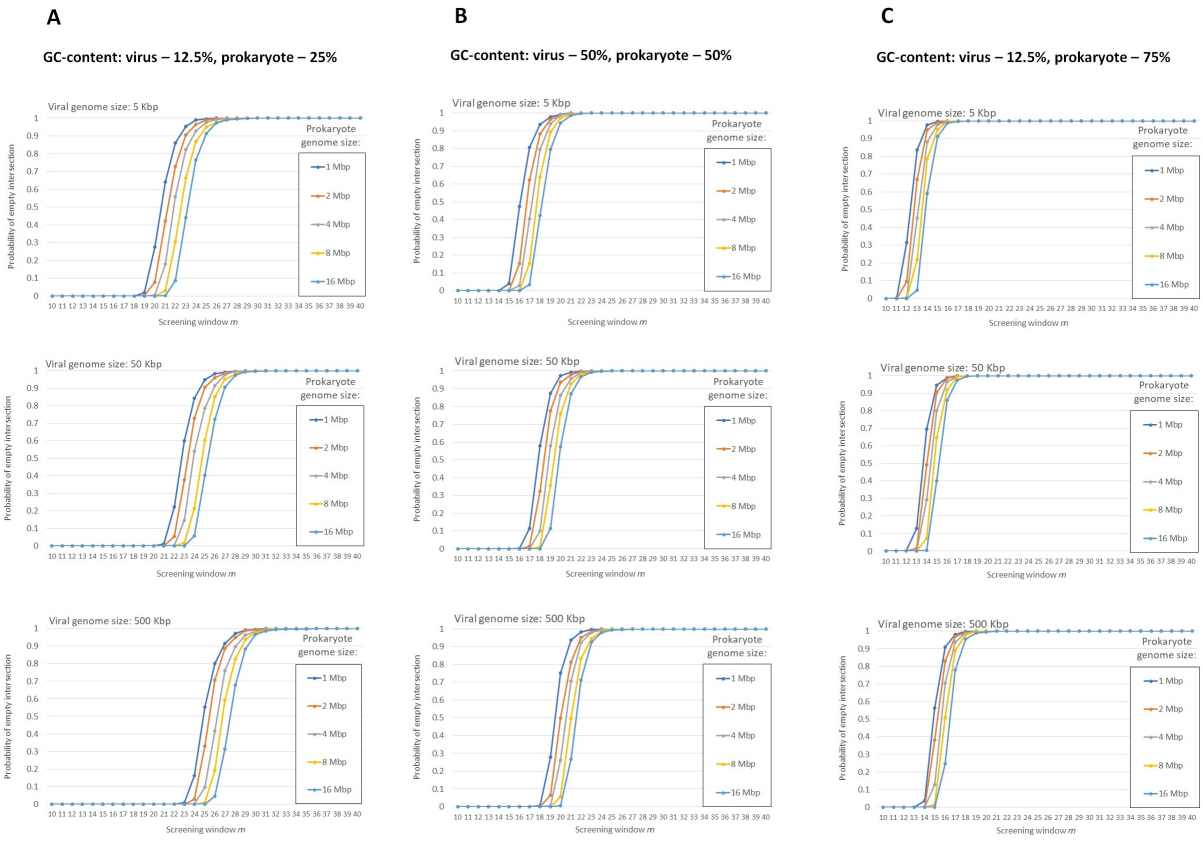

Figure 3.3: Simulation results for uniform distribution and possible extreme skewed cases for GC-content combinations: (A) phage – 12.5%, bacterium – 25%; (B) phage – 50%, bacterium – 50%; (C) phage – 12.5%, bacterium – 75% .

3.4 Comparison between theoretical and simulation estimates

The theoretically derived formula (Section 3.1) provides an opportunity to quickly estimate the probability of observing no overlap between genomes for a given value of m. The use of simulation for estimating the probability is more computationally intensive, and the statistical significance of this estimate depends on the number of simulations.

Figure 3.4 shows theoretical and simulation estimate for a pair of genomes, *E.coli K-12* and *Phage PhiX174*, shown earlier in Chapter 2 in Figure 2.3. Figure 3.4 demonstrate that the formula takes more conservative approach in estimating the probability, i.e., for a given value of m during the transition stage, the probability computed using the formula is slightly less than the probability computed using the simulation. For example, for $m = 19$, the theoretically computed probability equals 0.9268 and the probability computed using the simulation with 1024 runs equals 0.9156. We suggest the difference is likely due to the sliding window effect that is not taken into account in the formula.

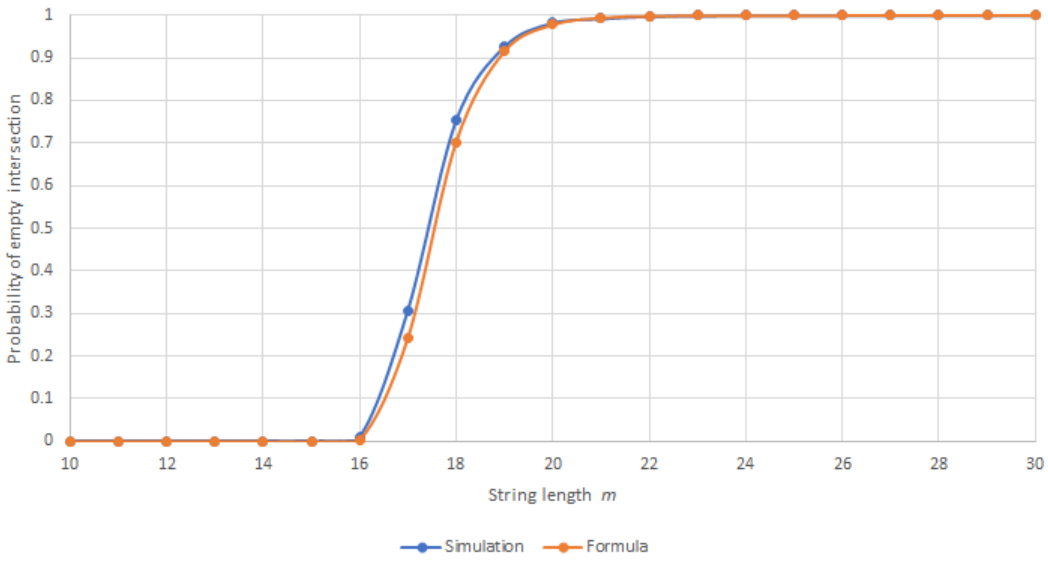

Figure 3.4: Comparison between the theoretical and simulation estimates for *E.coli K-12 MG1655* (NC_000913, 4,641,652 bp, GC 50.79%) and Coliphage phi-X174 (NC_001422, 5,386 bp, GC 44.76%) using simulated genomes of the same size and GC-content as genomes of the quoted organisms.

3.5 Conclusion

The developed theoretical and computational frameworks allowed us to estimate suitable screening window for various applications. Formula provides a computationally inexpensive and quick way to make an estimate. The estimate can be further adjusted and refined by means of statistical modeling.

Chapter 4

Viruses as probes for functional properties of their hosts

4.1 Abstract

Finding shared fragments between genomes is important for solving many biological challenges. Such fragments in microbial genomes suggest interactions between host bacteria and viral parasites helping to identify host-parasite associations (see Chapter 6) and answer other critical questions about the organisms involved. Current methods can be supplemented by new computational technologies for versatile analysis of unannotated genomic string interactions. The goal of this study is to determine statistically significant genomic intersections that can imply important biological meaning. We explore how these intersections can be used to predict pathogenicity and distinguish between different *Escherichia coli* (*E.coli*) strains.

We show the feasibility and usefulness of scalable computational algorithms to find pairs of organisms that interact with each other, such as bacterium-phage or host-parasite pairs, based on collected unannotated genome data. The statistical significance of the occurrence of matching strings is used to filter out matches possible due to chance. Our method, supplemented with machine learning techniques, can predict pathogenicity of bacterial strains using phage screening and profiling based

on sequenced genomes without the need of annotation. We applied the algorithms to find "fingerprint" of phages interacting with bacterial hosts by analyzing 2,480 phage genomes from European Nucleotide Archive (ENA). The methods have adjustable sensitivity and specificity in identifying phages and provide bacterial "fingerprints" in terms of phage presence in microbial genomes with the desired level of resolution for evaluating pathogenicity of *E.coli* strains. This chapter is heavily based on our publication (Lenskaia and Boley, 2018).

4.2 Background

4.2.1 Pathogenicity Prediction

E.coli strains come in many varieties. It can be a commensal bacterium that is a part of normal human intestinal microflora (Huttenhower et al., 2012) or it can be highly pathogenic and cause severe infections in animals and humans (Kaper et al., 2004). It is also one of the most well-studied microbes in laboratory settings (Escherich, 1988; Raetz, 1996; Dunne et al., 2017). It is an important bacterium in biotechnology that can produce insulin (Goeddel et al., 1979), biodiesel (Kalscheuer et al., 2006), and other compounds. Assessment of pathogenicity is very important for epidemiological (Rangel et al., 2005; Grad et al., 2012), food safety (Besser et al., 1993; Scallan et al., 2011), veterinary (Blanco et al., 2001), and other health-related studies. As the cost of whole genome sequencing decreases, the availability of complete sequence genomes increases rapidly. The ability to quickly estimate a strain potential pathogenicity based just on its raw genome sequence would be an important advantage in diagnostics. It would save time and resources that otherwise would be necessary for wet-lab experiments and other resource-consuming techniques like multiple alignments.

The evolutionary transition to pathogenicity can result from acquiring differ-

ent virulence factors when new genes responsible for producing toxins and other pathogenic components are incorporated into bacterial genomes. Although it is easy to determine presence of known genes in newly sequenced bacteria, some genes remain as unknown function. Bartoszek et al. (2018) created a model that was able to trace virulence factors based on persistence of trinucleotide repeats within clinical isolates.

However, the presence of virulence genes itself does not necessarily result in pathogenicity (Wassenaar and Gunzer, 2015). Bacteriophages (phage for short) are known for their contribution to the pathogenicity of bacteria (Penadés et al., 2015) as well as to adaptive traits and diversification of bacterial strains. Since their survival depends on their success in infecting bacteria and getting viable progeny, phages must find and examine every possible flaw in a bacterial cell. By exploring existing remnants from many phages in bacterial genomes, we can evaluate the overall picture of the bacterial genome state. Touchon et al. (2016) investigated associations of genetic and life-history traits in bacteria with the distribution of prophages. They found slight correlation between pathogenicity and the number of prophages across different species. Presence of different types of pathogenicity patterns may blur the overall picture across different species of bacteria (Brüssow et al., 2004). In this research, we focus on evaluating the contribution of prophages to life traits within *E.coli* strains and exploring its predictive power on potential pathogenicity of individual strains of *E.coli*.

4.2.2 Identification of shared fragments between host and parasite genomes

Genomes of different organisms might share extensive string fragments due to some biological reasons, e.g. temperate phages (Howard-Varona et al., 2017), prophages and phage remnants (Touchon et al., 2016). Long shared fragments are a sign of

a biological relationship between a host organism and a parasite. This may be a sign of an attack mechanism on the part of the parasite or a defense mechanism on the part of the host. By extracting and identifying shared genetic sequence fragments from bacteria and viruses, one can quickly distinguish between similar bacteria based on their functional behavior in the presence of phages, or quickly identify which viruses might be suitable as vectors with which to attack and destroy bacteria (phage therapy). This can also provide evidence of the historical evolution of bacteria and associated viruses. CRISPR-Cas defenses and other mechanisms based on recognizing substrings in viral genomes on the part of bacteria, and the mechanisms used by viruses to avoid recognition are still under investigation (Arber, 1978; Barrangou and Van Der Oost, 2013). Identification of long genome fragments is a challenging task. Existing methods for detecting shared fragments between host and parasite genomes can be roughly divided into two categories: 1) "wet lab" in-vitro microbiological methods (hybridization capture sequencing, microarrays,etc.) (Kim et al., 2012) and 2) "dry lab" in-silico computational methods (searching for the longest common substring, alignments, etc.). "Wet lab" methods are technologically intensive, timeconsuming, and have limited throughput. They work based on hybridization by detecting complementary base pairs between short segments of host genome and viral fragments (e.g. between human genome and retroviruses (Escalera-Zamudio and Greenwood, 2016), koala genome and retroviruses (Tsangaras et al., 2014)). These methods often depend on the use of specific primers to locate and amplify target fragments. "Dry lab" methods are computational and hence often very scalable and flexible. Computational approaches have been often successfully applied to analyze and distinguish between biological sequences (Grau et al., 2012). Currently, the main *in-silico* sources of information about phage incorporation into microbial genomes are annotated databases and software that make use of the annotations to identify bacteria-phage pairs. Examples include (1) special databases, e.g.,

ACLAME database (Leplae et al., 2004) and PhAnToMe http://www.phantome.org and (2) computational tools that depend on annotated databases, e.g. Phage_Finder (Fouts, 2006), Phaster (Arndt et al., 2016), and VirSorter (Roux et al., 2015). Many existing computational methods for bacteria-phage interaction depend on meta-data (e.g., coding regions, protein sequences, etc.) and external software solutions for localization of prophage regions, e.g. FASTA33 (Pearson, 1990), NCBI BLASTALL (Altschul et al., 1997), HMMSEARCH (Eddy, 1998), and MUMMER (Delcher et al., 1999). Unfortunately, the annotations are limited to those locations which have been explicitly identified to be of interest, making it difficult to identify possible new locations with unidentified segments.

An ever-growing quantity of genetic sequence data is being accumulated that is yet to be annotated or which function is unknown. Existing annotations vary depending on goals of individual databases. Attempts to standardize annotation exist, such as NCBI Prokaryotic Genome Annotation Pipeline, but annotations may change as new discoveries are made. Touchon et al. (2016) used Phage_Finder (Fouts, 2006) to predict phage incorporation. However, they mentioned that it was hard to distinguish phages and other mobile elements. To avoid ambiguity in identification of the origin of mobile elements, we use exact matching to known phage genomes. As reviewed by Edwards et al. (2016), substring matching is the most accurate method in terms of predicting host-parasite associations.

4.2.3 Exact matching for host-parasite interactions

We seek a computational screening method that works on raw genome assemblies and gives a common picture of candidate string interactions, without using any meta-data annotations. Such a method could be used on newly discovered bacterial variants, or on genome regions of unknown function. It needs to be scalable, sensitive to capture many interesting interactions, while specific enough to avoid false positives.

The method of "all common subsequences" (ACS) (Wang, 2007) is a computationally effective method that measures similarity relationship between sequences by extracting many common subsequences. Because the subsequences are not necessarily contiguous, this can suffer from ambiguities similar to those found in alignments. Although improved alignment methods are effective (Morgenstern, 1999) and currently developed alignment methods are very efficient (Al-Ghalith and Knights, 2017) https://github.com/knights-lab/BURST, the statistical significance of finding a particular pattern with this method is not trivial to estimate.

Seeking "all common [contiguous] substrings" avoids this alignment ambiguity and can be implemented efficiently using suffix trees and string kernels. Leslie et al. (2001) used a string kernel based on counts of short common substrings. Here we use a similar search for common substrings but consider much longer substring lengths to distinguish between biologically related and unrelated genomes.

4.2.4 Adjustment of sensitivity and specificity

Many newly created methods for detecting host-parasite associations based on string content demonstrate good performance. An in-depth review of existing methods to find host-parasite associations was done by (Edwards et al., 2016). Many such methods have been trained on specific datasets and locally optimized, however they sometimes suffer from a lack of specificity for large-scale screening research since they depend on statistical models on short substrings. For example, HostPhinder (Villarroel et al., 2016) predicts based on 16-mers; WIsH (Galiez et al., 2017) uses Markov models of order 8; the method developed by Zhang et al. (2017) uses frequencies of 6-lettered words, and VirFinder (Ren et al., 2017) utilizes 8-mers. Although this lack of specificity can be compensated to some extent by considering additional factors (annotation, metadata, biological knowledge, etc.) to distinguish from true biologically relevant and random (biologically nonrelevant) matches, it substantially limits

the ability of tools to work on raw genomic data *en masse*.

We compute the common substrings for a variety of fixed lengths between a host genome and a phage genome. The lengths chosen are long enough so that unrelated organisms are very unlikely to show commonality, while short enough to occur often among biological organisms of interest (see Chapter 3). The use of fixed length strings makes it easy to get good estimates of the statistical significance of the computed results based on simulation of a simple statistical model. Our computational strategy leads to a screening technique for fast and resource-effective preliminary analysis of host-parasite interactions in unannotated databases of complete genomes. It also allows the pairing of a given new bacterium with many phages to produce a sort of fingerprint for the bacterium, permitting rapid identification of new bacteria based on their functional interactions with phages.

The strings lengths can be adjusted to yield a variety of levels of resolution, sensitivity, and specificity in the results. These computational methods yield important information about statistically significant intersections between bacterial and viral genomes. These data can be analyzed by machine learning techniques to obtain important patterns of phage contribution to properties of bacterial strains. We hypothesize that it is possible to estimate host pathogenicity based on genetic sequence overlap with a library of phages. We develop an efficient computational tool to test this hypothesis and validate it on a set of *E.coli* strains and associated phages. Our algorithms have been implemented in Python. For an individual host, the measures of overlap with a large collection of phages can be considered as a sort of functional "fingerprint" of the bacterial host, which can be assembled from raw genome sequence data. In this study, we demonstrate that the fingerprints (assembly of interaction levels with many phages) can distinguish between benign and pathogenic strains of *E.coli* and that these methods are then followed by machine learning can be used to predict pathogenicity in bacteria.

4.3 Methods

For the input, the algorithm takes complete genome sequences in FASTA format. During our analysis, it assembles the dictionary of strings representing each individual organism and then computes indices of pairwise overlap between each pair of organisms in question. These indices are then used as predictors of certain functional properties of bacterial hosts, specifically their pathogenicity. As results, the algorithm produces a classifier and returns classification results. These results are used to identify a list of phages that are most capable to distinguish between pathogenic and other strains. These indicator phages allow to reduce feature space and increase the classifier's accuracy.

Experimental procedure steps:

1. Choose appropriate string length m based on statistical simulations
2. Construct phage fingerprints (pairwise indices between a host and phages)
   (a) Assemble dictionary of all substrings of length m for each given raw genome
   (b) Compute intersection indices between bacterial and phage dictionaries
3. Apply machine learning classifiers
   (a) Divide the dataset into train and test sets for 10 fold cross-validation
   (b) Train classifiers
   (c) Test classifiers
4. Determine a set of "indicator" phages

4.3.1 Dictionary assembly and computation of phage fingerprints

As an input file, the algorithm takes unannotated genomes in FASTA format. For a given genome G, we obtain its dictionary $D_m(G)$ by scanning the genome with a sliding window of length m, sequentially shifting it one nucleotide at a time. The results are assembled into a table of unique strings ("keys"). The dictionary $D_m(G)$ consists of all distinct contiguous substrings of length m present in G. The size of the dictionary equals the number of distinct substrings. We could also store the number of occurrences of each substring, but this information was not used in the computations reported here.

To find strings of a given length shared between two genomes (H – host, P – parasite), we need to compute a measure of the degree of intersection between two sets of dictionary keys by filtering out entries present in both dictionaries. We define the intersection ratio of parasite genome P within host genome H to be the number of distinct common substrings of length m divided by the size of P's dictionary (see Chapter 2, equation (2.3)). For each bacterial host, we computed a phage fingerprint by calculating the intersection ratios between the host and a collection of phages.

4.3.2 Determine appropriate window size

We use a simple statistical model to determine the range of appropriate window sizes. An appropriate window size must be long enough to avoid string overlaps by pure random chance [specificity], but short enough to capture relations between organisms [sensitivity].

Statistical modeling to determine appropriate window size

We first estimate the probability of obtaining a non-empty intersection between two random genomes. To obtain estimates of the occurrence of non-empty intersections by chance, we repeatedly simulate the generation of dictionaries from randomly generated "genomes" 1024 times using IID with uniform distribution of nucleotides. This process is carried out for each length n of interest. Since the substrings of length m arise from a sliding window, they are not statistically independent, so they cannot be modelled by a simple statistical distribution over independent individual bps, like a multinomial distribution. Hence we use a numerical simulation.

According to the results of numerical simulation, for any value of m up to 16, there is always some entry in the intersection. For any value of $m \geq 25$, the observed probability of anything in the intersection is no greater than 0.0001. This gives us a threshold for a non-specific area. For values of m in a range from 17 to 24, intersection may be present or absent. The observed thresholds remain invariant (stable) within the range of analyzed *E.coli* genomes (4-6 Mbp) and viral genomes (2-500Kbp) (Figure 4.1) even as the GC content of the latter varied from 25% to 75%. According to the results of computational experiments, GC-content of phage genome has little effect on shifting the threshold. *E.coli* strains have GC-content close to 50% and the distribution of single nucleotide within *E.coli* genome is close to uniform. Thus, to make computations as simple as possible, we use a uniform distribution of nucleotides for both bacterial and viral genome to model the corresponding intersections.

Screening sensitivity and specificity

To demonstrate how sensitivity of the screening method varies as a function of string length, we counted the number of phages having non-empty intersection with two representative hosts, one for each class, *E.coli O157:H7 Sakai* (BA000007) for pathogenic

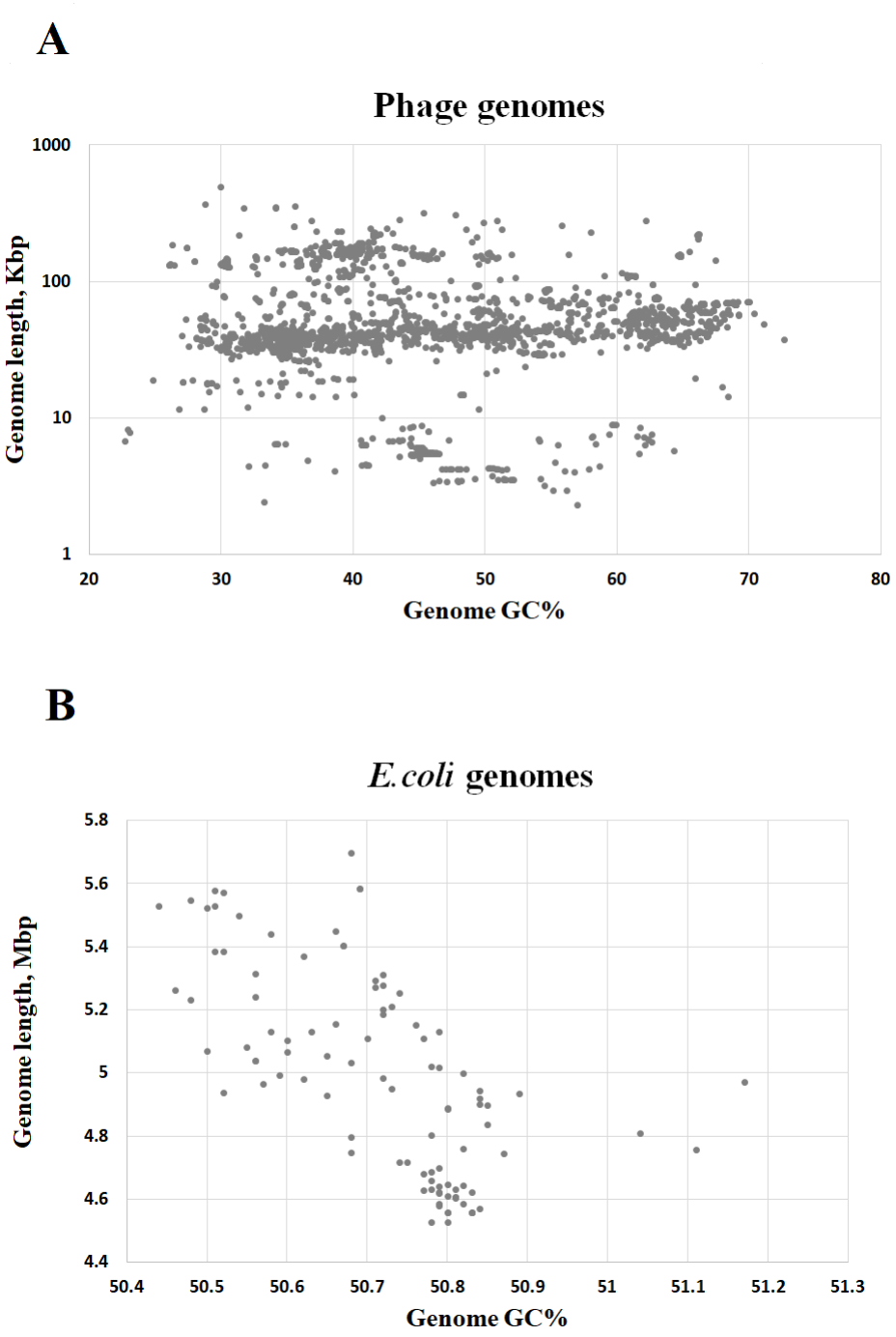

Figure 4.1: These diagrams represent the distribution of lengths andGC% for the analyzed genomes: (A) Phages; (B) *E.coli* strains.

strains and *E.coli K-12 MG1655* (NC_000913) for other strains, for different string lengths m (Figure 4.2). There are three areas: (I) nonspecific area with high sensitivity; (II) "stable" sensitive and specific area; (III) specific area with low sensitivity. The string length must be above 25 bp to distinguish from random, but over 50 bp tends to lose sensitivity to some phages, hence the choice of $m = 40$ bp is appropriate. To evaluate specificity of our method we screened the two strains of *E.coli* against 4,743 viruses of eukaryotes (plant, animal, human, etc.) available in ENA http://www.ebi.ac.uk/ genomes/virus.html using the same string length 40 bp. We found only two viruses having nonempty intersection, albeit with small intersection ratio values: *Vaccinia virus GLV-1h68* (EU410304) with *E.coli K12* (intersection ratio .008825) and with *E.coli O157:H7* (intersection ratio .005842); and *Cyprinid herpesvirus 1 strain NG-J1* (JQ815363) with just *E.coli O157:H7* (intersection ratio .000024), in all cases with small intersection ratio values. The presence of common strings of length as long as 40 bp even in such small amounts between these eukaryotes' viruses and *E.coli* strains is very interesting and deserves further investigation. Such a small number (2 out of 4,743) of false positives for string length 40 bp is a high degree of specificity for the developed method.

The choice of string length m represents a trade-off between sensitivity and specificity and depends on the specific genomes under study. Filtering phages based on intersections computed with multiple values of n might be appropriate for different host-parasite pairs with genomes of different lengths. This is a direction for future investigation. This choice of $m = 40$ bp provided a suitable balance between sensitivity and specificity for *E.coli* and associated phages. Our goal is to identify possible interactions of phages within host bacteria beyond simple defense mechanisms. Hence, we avoid false positives by choosing lengths above the typical spacer length in CRISPR-Cas loci of the bacterial hosts. According to the information retrieved from CRISPRdb http://crispr.i2bc.paris-saclay.fr/ accessed on May 8, 2018 more than 90%

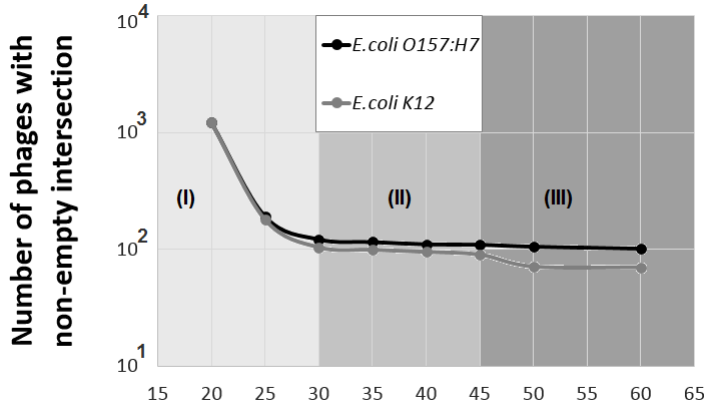

Figure 4.2: Number of phages with non-empty intersection for *E.coli O157:H7* and *E.coli K12* for different string lengths: (I) high sensitivity; (II)" sensitivity plateau"; (III) decreasing sensitivity. According to the results of statistical modeling, area (I) is nonspecific with high number of false positives; areas (II) and (III) are specific.

of known spacers in microorganisms have length below 40 bp.

4.3.3 Data preparation

We computed the intersection ratios ($m = 40 > 25$ bp) for 2480 phage genomes with respect to 101 *E.coli* genomes available in ENA to obtain phage fingerprints for each bacterial host. This allowed us to compare phage contribution to *E.coli* genome and their impact on pathogenicity of different strains. We kept only those phages that had the nonempty statistically significant intersection with *E.coli* genomes. We found 172 phages that had their fragments inserted in at least one *E.coli* genome of interest. Within each *E.coli* genome, we found remnants of no less than 30 phages. Maximum number of phages which remnants were identified within a genome of sequenced *E.coli* strain using our screening method was 127. The phage fingerprints were used as feature vectors for classification of bacterial hosts by machine learning methods into pathogenic and non-pathogenic strains. For each strain, we obtained information about its pathogenicity from the literature. We treat specific strain as potentially

pathogenic if they were indicated to cause infection in animals or humans. Other strains include biotechnological strains, commensal strains obtained from healthy individuals, and laboratory strains.

4.3.4 Machine learning classifiers

We apply machine learning methods to investigate the possibility of inferring pathogenicity based on phage fingerprints. We use random forests (Breiman and Cutler, 2007) as a classifier since this algorithm has embedded feature selection, keeps only important features, and handles dimensionality well. It takes a bootstrap sample from the data and fits a classification tree. At each node, it randomly selects a number of features, i.e., *mtry* parameter, from all features in the data, finds the best possible split considering this number of features, and grows the tree further. It uses voting for determining the best decision path based on the constructed trees. It provides out-of-bag (OOB) error to estimate the generalization error and evaluate future performance. OOB is computed based on analysis of a confusion matrix using permutation of features.

We used randomForest (Breiman, 2018) and caret (Kuhn, 2018) packages in R. As input, the algorithm used phage profiles for 101 sequenced *E.coli* strains computed by our algorithm. A phage profile is a vector that stores pairwise intersection ratio values for a phage genome and each of the selected E.coli genomes. Thus, the size of each feature vector is (101x1) and we have 172 such vectors (one for each phage). It constitutes our set of predictors. For each *E.coli* strain we have a pathogenicity label which takes value of 1 for pathogenic strains and 0 for other strains (commensal, laboratory, biotechnological).

The random forest classifier predicts the pathogenicity of a bacterial strain based on the fingerprint of phage remnants in its genome. To avoid overfitting and get a reasonable estimate of model performance on phage profile data, we used 10-fold

cross-validation. We tried different cut-off for the m parameter, the number of features at each split, including the default value equal to the square root of the total number of features ($mtry = 13$), half the default value ($mtry = 6$), twice the default value ($mtry = 26$), and the total number of features ($mtry = 172$), to evaluate the relationship between prediction accuracy and the number of features necessary and sufficient to do the effective separation without overfitting.

To evaluate the contribution of features to the purity of separation on each step, the random forest algorithm (Breiman, 2018) can compute the mean decrease in Gini coefficient closely related to AUC (Hand and Till, 2001) as a measure of information gain. We use the average value of this measure in 10 folds to get the list of phages arrange in decreasing order of their importance with respect to the purity of separation between pathogenic and other *E.coli* strains. The number of folds were experimentally determined as the one that provided good estimates for model parameters on this data. We set different cut-offs for this list to determine the critical number of phages sufficient for proper classification of *E.coli* strains. Then we rebuild a prediction model on this reduced set of features and evaluate prediction accuracy. Finally, we used the cut-off that provides the highest level of accuracy as a reasonable estimate for the number of "indicator" phages.

4.4 Results

4.4.1 Screening window size

We found a threshold on the string length m ($p < 0.001$ where p is a probability of finding non-empty intersection between host and parasite genomes by chance) to distinguish between random and biologically related shared fragments for genomes. For the reported strains of *E.coli*, this threshold equals 25 bp. Thus, we find a range

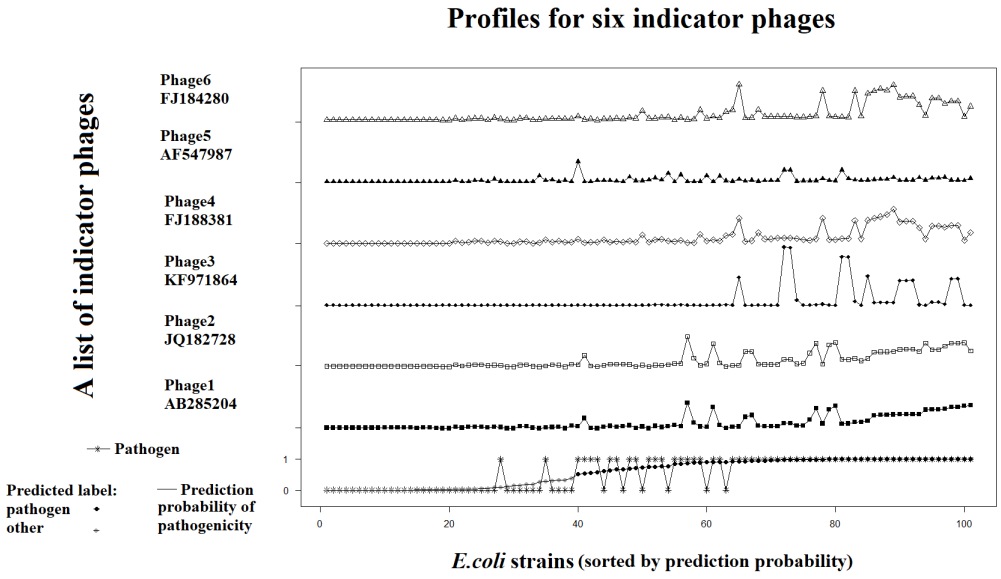

Figure 4.3: Profiles for the top six phages and the results of random forests prediction on this reduced set of features. The *E.coli* strains have been sorted by predicted probability of pathogenicity (the bottom stripe), based on these 6 phages. The bottom stripe also shows the "true" pathogenicity. The remaining stripes show the intersection ratio values of each phage across the 101 *E.coli* strains.

of lengths starting at the threshold and extending to the length of a phage genome (at maximum) that allows us to analyze biologically important intersections between host and parasite genomes. We can vary the string length in this range for screening to obtain a desired level of specificity and sensitivity while analyzing shared fragments between genomes.

4.4.2 Prediction of functional properties: Pathogenicity

To estimate a possible difference in phage remnants between pathogenic and other strains of *E.coli*, we investigated two well-studied representatives of *E.coli* with available reference genomes: pathogenic – *E.coli O157:H7 Sakai*, benign – *E.coli K-12 MG1655*.

Figure 4.3 shows the phage presence in these two strains. We found 115 phages

that have non-empty intersection with at least one of the two selected strains. Interestingly, we found that 91 of 115 (80%) phages were common for both bacteria. Spearman's correlation for the contributions of common phages between these two bacteria is 0.6 which suggest a strong positive correlation. The remaining 24 of 115 (20%) of phages were present only in one of the two bacteria (20 in *O157:H7* only, 4 in *K12* only).

Although the lists of found shared phage components were very similar, the amount of actual insertion for common phages were significantly different between the two strains. To estimate the difference between these amounts, we create distance metrics based on the sum of absolute differences between the intersection ratio values. On average, the pathogenic strain of *E.coli* has 60 times large values of the intersection ration for common phages. This observation suggests positive association between pathogenicity and the amount of phage remnants within the host genome. We then investigated its predictive power over the entire set of bacterial hosts by machine learning. Based on the computed overlaps between the 101 *E.coli* strains with the 172 phages (ignoring the 2308 phages that showed no overlaps at all), the random forest classifier yielded an average out-of-bag error rate in 10 folds of 12.84% $\pm$ 1.67%. The best average accuracy in 10 folds equaled 89.21% $\pm$ 10.68%, obtained with $mtry = 6$. The average accuracy in 10 folds across different $mtry$ values was 88.74% $\pm$ 0.04%.

4.4.3 Identification of most distinguishing individual phages

We seek a small number of phages that is sufficient to do a complete classification. We called it "indicator" phages. To better understand the importance of features in making a decision on splits, we used mean decrease in Gini measure. Based on constructed random forests in 10 folds, we formed a list of the most important phages used by the algorithm to distinguish between pathogenic and other strains of *E.coli*.

Considering the results for the random forests prediction model with the best accuracy achieved, we selected 6 most frequently used phages from this list. Then we reduced the list of features to those six phages and retrained the prediction model. The result is shown in Figure 4.3, with a slightly higher level of accuracy on the reduced set of features: 1) $91.94\% \pm 7.62\%$ (6 phages, $mtry = 3$); 2) $89.21\% \pm 10.68\%$ (all phages, $mtry = 6$).

The results indicate that 6 is a suitable number of features to separate pathogenic and other strains. The six identified phages have similar genome length (51.44 ± 8.56 Kbp) and GC% ($49.96\% \pm 1.30\%$). Five of the six phages belong to *Caudovirales*, they are dsDNA viruses: *Enterobacteria phage cdtI, Shigella phage Sf6, Stx2-converting phage 1717, Enterobacteria phage mEp460, Escherichia phage phi191.* The remaining virus is an unclassified bacterial virus: *Enterobacteria phage YYZ-2008.* It is worth noting that three of the identified phages have zero intersection ($m = 40$) between each other indicating their mutual orthogonality in feature space. The remaining three phages have overlaps that indicates their similarity. However, the existing differences between them make sufficient contribution to prediction accuracy (Figure 4.3). Profiles for the top six phages and the results of random forests prediction on this reduced set of features.

The *E.coli* strains have been sorted by predicted probability of pathogenicity (the bottom stripe), based on these 6 phages. The bottom stripe also shows the "true" pathogenicity. The remaining stripes show the intersection ratio values of each phage across the 101 *E.coli* strains. Phages having statistically significant intersection with *E.coli* can help to distinguish between pathogenic and other strains using machine learning. Relationships between phages and *E.coli* hosts and between phages themselves are non-linear. Some phages have synergism and some exhibit antagonism. However, the number of sequenced phage genomes is sufficient for machine learning search of indicator phages to predict pathogenicity. It is possible to find a combination

of phages among sequenced ones that provide a high prediction accuracy ($> 91\%$).

4.5 Discussion

4.5.1 Phage fingerprints

We have shown that automated algorithms based on analysis of long unique shared strings applied to unannotated genome data can yield useful information about bacteria-phages interactions. We use statistical modeling for searching and investigating significant similarities between genomes in string diversity. It was possible to find a threshold above which it is virtually impossible to find unique strings common to two different unrelated genomes. Finding such pairs above the threshold strongly suggests an existing biological or evolutionary relationship between these genomes. Further investigation is needed to determine thresholds for genomes from other organisms, and to combine the filtering results from multiple string lengths.

We have used the screening method to construct a functional "viral fingerprint" for *E.coli* strains, where each fingerprint distinguishes between strains based on their evolutionary relationship to a wide variety of phages. Our analysis revealed the entire range of interactions between bacterial and phage genomes: no incorporation, partial incorporation, and almost complete incorporation of phages into microbial genomes. For example (Figure 4.4), it was found that *Stx2 converting phage II* (AP005154) had the intersection ratio value value above 0.9976 for *E.coli O157:H7 Sakai* (BA000007) indicating almost complete incorporation of this phage into this host genome. We found that the intersection ratio values for the pathogenic strain *E.coli O157:H7 Sakai* are significantly higher than for the lab strain *E.coli K-12 MG1655*. It would suggest more active interactions between phages and pathogenic strains than laboratory strains.

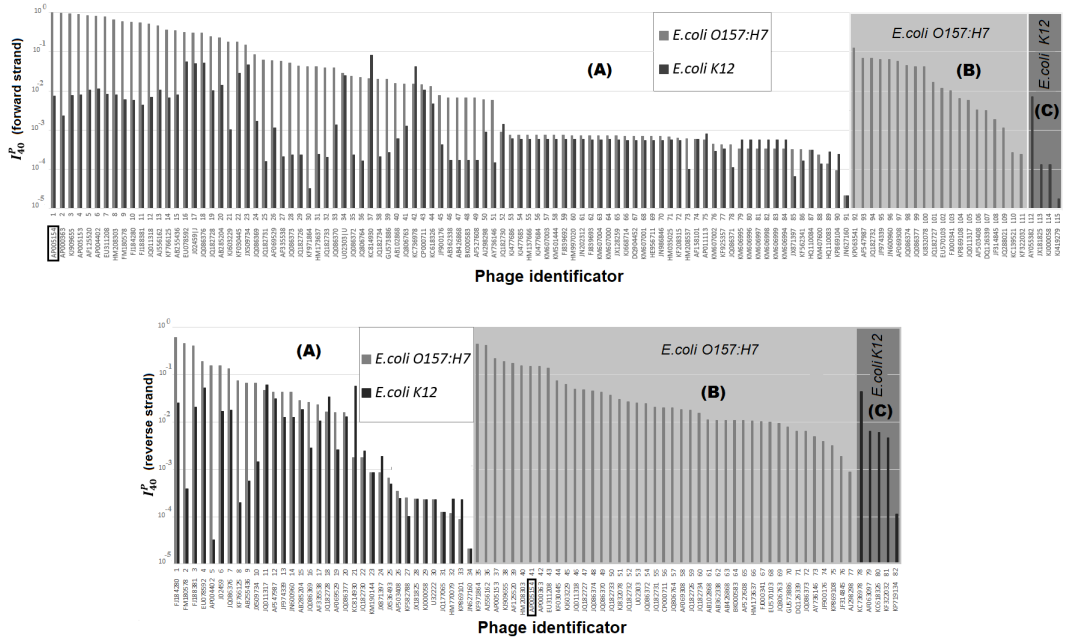

Figure 4.4: Among 2,480 phages in ENA, 115 phages have non-empty intersection with *E.coli O157:H7 Sakai* (BA000007) or *E.coli K-12 MG1655* (NC_000913) on forward strand: 91 phage have common strings with both strains (area A), 20 phages have common strings only with *E.coli O157:H7* (area B), and 4 phages have common strings only with *E.coli K-12* (area C). For reverse strand, 82 phages have non-empty intersection with *E.coli O157:H7* or *E.coli K-12*: 34 phage have common strings with both strains (area A), 43 phages have common strings only with *E.coli O157:H7* (area B), and 5 phages have common strings only with *E.coli K-12* (area C). Phages are arranged by decreasing order of the intersection ratio values with *E.coli O157:H7* (areas A and B) and in *E.coli K-12* (area C). Black frames indicate a position of *Stx2 converting phage II* (AP005154) that is almost completely incorporated into *E.coli O157:H7* genome.

This might help in predicting pathogenicity of newly sequenced strains of *E.coli* based on phage occupancy of their genomes. The "wildness" of a bacterial strain (history of exposures to varied environments) might be predicted by a high degree of interaction with a variety of viruses, as indicated by high degree of virus incorporation. This warrants further investigation, including the possible use of overlap occurrence counts (not used in the present analysis).

Moreover, the collected indices of phages incorporated into bacteria can be considered as a fingerprint for the bacteria (Figure 4.4) in order to (1) classify distant strains with the help of common phages from area (A); (2) identify and distinguish between closely related strains using the differences in their phage indices between areas (B) and (C). The differences between (B) and (C) areas also can be applied for microbial typing as alternative to typing based on CRISPR loci (Briner and Barrangou, 2014).

Currently there is a trend in transiting from "wet lab" to "dry lab" methods to carry out voluminous tasks such as epidemiological studies (Chattaway et al., 2017). Based on screening results, this method can locate the most significant area(s) for fingerprints to fulfill specific research purposes. The potential range of capabilities for the algorithms proposed in this paper are limited primarily by the presence of virus and bacterial sequence data within databases.

In addition, such fingerprints allow us screen for phages with potentially high level of similarity (bars of equal length on the forward strand fingerprint in area (A), Figure 4.4). Such phages deserve a close look at their mutual similarity. We can sort out these viruses based on the screening results and investigate the relationship in detail using similarity screening and alignment methods. The double impact of phages with resembling levels of similarities could be downweighted or excluded, depending on the variability. However, certain level of differences in content between highly similar phages might provide an important typing advantage.

4.5.2 Pathogenicity prediction

We found that information about phage occupancy of host genomes is a good predictor of host potential pathogenicity. We applied machine learning techniques to predict pathogenicity of *E.coli* strains based on their phage spectra since currently databases contain sufficient amount of host and parasite genomes for this species. For each *E.coli* strain with sequenced genome available in ENA, we found at least 30 phages with sequenced genomes which fragments were identified in a host genome. With growing availability of other bacterial and viral sequenced genomes it is possible to expand this prediction approach to other species. The presence of long common substrings of over a hundred phages in the *E.coli* strains indicates a significant pressure of those viruses on the host. The presence or absence of common substrings, computed in an automated way, can be used to distinguish bacteria phages from other viruses, identify particular phages that could be used as vectors against a very selective group of bacteria strains, and to distinguish between superficially similar bacteria based on differences between their evolutionary history and/or their putative functional interaction with their "viral environment".

We found that six "indicator" viruses are sufficient to distinguish between pathogenic and other *E.coli* strains (Figure 4.3). Since bacteria and viruses adapt quickly and they are able to change their genome rapidly (mutations, horizontal transfer, etc.), the detected indicator viruses have the best predictive power in relation to the current state of the analyzed bacterium genomes. However, the described approach allows to identify a relevant set of indicator viruses for genomes placed in different time frames and environmental conditions. This method is capable to reveal indicator phages for distinguishing between potentially pathogenic and other strains. It also might help to locate current pathogenicity hot spots in *E.coli* genomes.

In conclusion, we observed the interconnection between phage occupation of *E.coli* genomes and potential strain pathogenicity. We applied this to develop a computa-

tional "dry-lab" technology to predict pathogenicity of *E.coli* strains using phage screening of their unannotated sequenced genomes. The accuracy of the method will only increase with growing availability of sequenced viral and bacterial genomes in the databases.

4.5.3 Method applications

The methods proposed here do not depend on annotations. Due to exact matching, they are very specific and able to detect and distinguish between even closely related phages. This approach allows to accurately identify integration of phages into host genomes, but it has limited ability to detect interaction without such integration. It is currently optimized to detect integration of phages into host chromosomes, but it could be adapted to other types of genetic integration. The computational complexity of the methods is linearly related to the size of analyzed genomes which is an important advantage for a screening tool. The methods here have potential in monitoring host-parasite interactions and tracking different trajectories of viral fragments inside microbial genomes: incorporation of certain fragments, further increment/decrement in a number of copies, and elimination of particular viral fragments. Currently our method works on unannotated complete genomes, but similar methods could potentially be developed to work on raw genome assemblies (scaffolds) and reads, to form a handy software screening tool for laboratory and medical applications, e.g. identifying prospective candidates for phage therapy and monitoring interactions between microbial and viral genomes during treatment.

4.6 Conclusion

Our approach allows us to detect a degree of viral incorporation into bacterial genomes with varied levels of resolution and with various goals. The resolution can vary in a

range of string lengths above the threshold obtained by analysis of shared strings and evaluation of findings by statistical modeling. Differences in the values of the intersection ratio suggest differences in the evolutionary history of genome interactions. For example, wild type *E.coli O157:H7* has generally higher values of the intersection ratio compared to artificial *E.coli K12* developed in a "sheltered" laboratory environment (Figure 4.4). In our opinion, this approach can be useful for exploring different functional states of genomes, e.g., pathogenicity, antibiotic resistance, virulence. The selectivity will only grow as the databases grow and the methods are applied to ever wider classes of genomes. The methods can be used to screen genome sequence data semi-autonomously without any annotations. It can be useful as an early screening tool to find potential new biological interactions or selective interactions, as precursor to more in-depth validation *in-silico* with other meta-data or *in-vitro*.

Chapter 5

The exploration of autoimmunity potential in prokaryotes

5.1 Abstract

We analyze risks of autoimmune reactions associated with CRISPR-Cas systems in prokaryotes by computational methods. We found important differences between Bacteria and Archaea with respect to manifestations of autoimmunity. According to the results of our analysis, CRISPR-Cas systems in Bacteria are more prone to selftargeting even though they possess several times less spacers per organism on average than Archaea. The results of our study provide opportunities to use self-targeting in prokaryote for biological and medical applications, e.g., for treatment of bacterial infections. This chapter appeared in (Lenskaia and Boley, 2020a).

5.2 Background

Adaptive immunity was first demonstrated in prokaryotes in 2007. Many important findings led to this discovery and helped put the puzzle of this enigmatic mechanism together (Ishino et al., 2018). In 2007, Barrangou et al. (2007) found experimental evidence of the hypothesized function of the segments consisting of repetitive structures

(spacers-repeats) and associated genes previously found in prokaryote genomes. Later practical protocols using these systems for precise genome editing attracted great attention (Pickar-Oliver and Gersbach, 2019). However, many questions regarding the fundamental mechanism of adaptive immunity still remain open.

The most poorly understood part of this immunity mechanism is spacer acquisition (McGinn and Marraffini, 2019). Criteria that bacteria use for spacer selection are under investigation (Nasko et al., 2019). Researchers suggest that molecular mechanisms can play a role in determining the size of spacers at least for some bacterial species (Nuñez et al., 2015). However, the question of spacer selection remains open given that bacteria utilize very rapid and extensive exchange of genetic materials (Dutta and Pan, 2002). All these findings raise a question of how self-targeting occurs in prokaryotes.

Stern et al. (2010) carried out the first systematic search of self-targeting spacers using the information from CRISPRdb (Grissa et al., 2007) about CRISPR structures found in the sequenced genomes available at that time. The authors found over a hundred of self-targeting spacers in 330 organisms (0.4% of 23550 spacers in total). After previous sketchy reports about the observed self-targeting events, that study provided the first systematic estimate of self-targeting rate in prokaryotes: "59 of 330 (18%) CRISPR-encoding organisms possess at least one array with at least one self-targeting spacer". Stern et al. also explored the hypothesis about a suggested role of self-targeting spacers in gene regulation and rejected it. Their conclusion was that self-targeting is a form of autoimmunity with a negative fitness cost. They also outlined possible ways to escape autoimmunity for prokaryotes including inactivation of self-targeting spacer, inactivation of CRISPR-Cas system, mutation of self-protospacer.

Subsequent researchers demonstrated that self-targeting spacers can be a marker of the presence of CRISPR-Cas inhibitors (Rauch et al., 2017; Watters et al., 2018). These inhibitors mostly encoded by phages represent anti-CRISPR mechanisms that

can help phages overcome CRISPR systems. These inhibitors may be used to control artificial CRISPR-Cas systems in the process of genome editing in eukaryotic cells.

These observations indicate that self-targeting events deserve further exploration. We use newly developed dictionary-based methods to facilitate this analysis. First, we repeat the initial analysis made by Stern et al. in 2010 to benchmark our methods. Second, we apply the same analysis to the current data available in CRISPRdb (3261 prokaryotes, 167,583 spacers). Our analysis aims to answer the following questions: (1) Are Archaea more prone to self-targeting compared to Bacteria? (2) Is there a difference in spacer length with respect to self-targeting between Bacteria and Archaea; (3) Are self-targeting spacers more often located on plasmids than on chromosomes? The answers to these questions help us to better understand self-targeting mechanism in the context of our current knowledge about CRISPR-Cas systems. In turn, it will provide opportunities to utilize self-targeting for biological and medical applications.

5.3 Methods

Information on the found CRISPR structures was obtained from CRISPRdb (the latest update, May 9, 2017). We downloaded the xml file for all analyzed prokaryotes that have at least one confirmed CRISPR array. Based on these data, we have compiled a list of organisms in the genomes of which CRISPR structures were detected. We downloaded the genomes of these organisms from the NCBI Nucleotide database. If genomes of organisms contained several replicons (i.e., chromosomes and plasmids), then each replicon was analyzed separately, and then the results were summarized at the organism level.

We extracted information about 330 organisms analyzed by Stern et al. As a reference, we used the list of the analyzed CRISPR arrays and the list of found self-targeting spacers provided by Stern et al. in the supplementary materials. We

re-analyzed these data, using the most current information available at CRISPRdb. Stern et al. searched for self-targeting spacers using BLAST alignment (Altschul et al., 1997) with a high similarity threshold to find 100% identity matches. However, many self-targeting events found by Stern et al. and included in the list of self-targeting spacers did not contain information about polarity. Moreover, BLAST utilizes a heuristic approach; it does not guarantee the search for all possible solutions. Instead of using an alignment-based approach, we use an "exact matching" approach inspired by the CRISPR mechanism itself. To search for exact matches and to accurately determine the polarity, we utilized our dictionary methods.

This approach is made efficient by using a dictionary (hash table) data structure. To search for self-targeting spacers in the genome of prokaryotes, we took information about all found CRISPR structures. We grouped all the found spacers by length. For each of their possible lengths, we compiled a dictionary with the unique strings of a given length as the keys and the lists of positions of these strings in the genome as the values. Then, we searched the dictionary for all spacers of that given length. To find copies on the forward and reverse strands, we searched the dictionary for the spacer (copies of the spacer on the direct strand) and its reverse complement (copies of the spacer on the reverse strand). As a result, for each spacer, we recorded into the output file its content, length, its position in the sequence and position(s) of the found copies of this spacer on the forward and reverse stands in the sequence. This helped us accurately identify the localization and polarity for self-targeting spacers. Then we compared all the found self-targeting spacers to those reported by Stern et al. Next, we conducted a similar analysis on all the data currently available at CRISPRdb. Later, we applied our analysis (Lenskaia and Boley, 2020a) to CRISPRCasdb dataset (Pourcel et al., 2020) that contained a larger collection of organisms (6865 prokaryotes, 221,397 spacers). The core methods and results are available on GitHub: https://github.com/tlensk/Self-targetCRISPR.

Table 5.1: The comparison between the results reported by Stern et al. and the results of the analysis using our dictionary-based methods.

Category	Number	%
1. Matched position(s) of self-targeting events	113	97.41%
2. No self-targeting events within the analyzed genome	3	2.59%
Total	116	100.00%

5.4 Results

We benchmarked our dictionary-based methods for the analysis of self-targeting events on the dataset of Stern et al. In 97% of the cases (113 of 116 spacers found by Stern et al.), the number and positions of self-targeting events in organisms coincided between the events reported by Stern et al. and identified by our methods (Table 5.1). The remaining three spacers belonged to one CRISPR array (NC_010125_1) and were previously reported by Stern et al. as self-targeting events when the CRISPR array bearing the self-targeting spacers was located on one genomic sequence (NC_010125) and their potential targets were captured on another genomic sequence (NC_011365). We found that both sequence accession numbers refer to a single chromosome in organisms that belong to the same bacterial species (*Gluconacetobacter diazotrophicus PAl 5*). Those three previously reported spacers most likely represent false hits when replicons from two distinct, but closely related genomes were jointly analyzed as if they belonged to a single organism in CRISPRdb, and this ˚ow inherently persisted during the previous analysis. This issue was resolved in CRISPRCasdb. The result of this comparative analysis validated the dictionary method as a viable approach to identify self-targeting spacers.

Table 5.2: The number of organisms with self-targeting spacers in Bacteria and Archaea.

Organisms	Bacteria	Archaea
Self-targeting	892	65
No self-targeting	2166	138

Table 5.3: The number of self-targeting spacers in Bacteria and Archaea.

Spacers	Bacteria	Archaea
Self-targeting	2325	163
No self-targeting	137514	27581

Having validated our dictionary approach, we applied it to all the spacers currently stored in CRISPRdb. We analyzed 3261 prokaryotes of which 957 (29.35%) have self-targeting spacers (Table 5.2). We found 2488 self-targeting spacers, approximately 1.5% of all 167,581 spacers (Table 5.3).

5.4.1 The comparison of self-targeting spacer rates in Bacteria and Archaea

There is a significant difference between Bacteria and Archaea with respect to the rate of self-targeting spacers (Table 5.3, Chi squared test, $p < 2.2e - 16$). The rate of self-targeting spacers in Archaea is (0.59%) is lower than the rate of self-targeting spacers in Bacteria (1.66%).

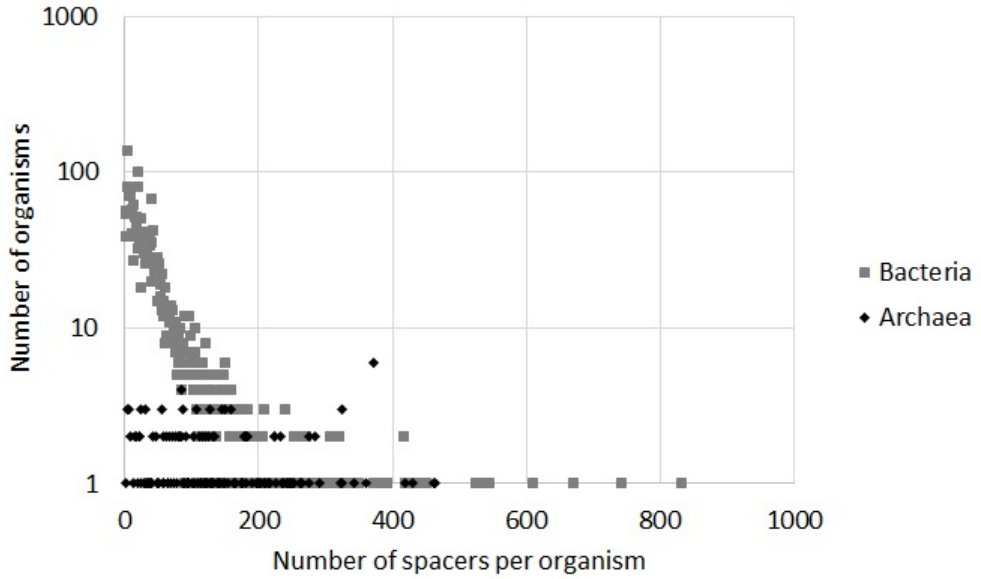

Figure 5.1: The number of spacers per organism for Archaea and Bacteria. The distributions are plotted using the semi-log scale.

The comparison of the distributions of the number of spacers per organism in Archaea and Bacteria demonstrates that these distributions are quite different (Figure 5.1). For Bacteria, most organisms have less than 50 spacers with the median of 28 spacers; for Archaea, most organisms have 100-150 spacers with the median of 116 spacers.

5.4.2 The spread of self-targeting spacers in Archaea

We found that 163 self-targeting spacers were spread across 65 of 203 (32.02%) archaeal organisms (Figure 5.2A). The organisms contained from 1 up to 20 self-targeting spacers. More than half of the organisms with self-targeting spacers, 34 of 65 organisms (52.3%) had only one self-targeting spacer. Another 16 organisms (24.62%) had exactly 2 self-targeting spacers. The remaining 15 organisms (23.08%) had from 3 to 20 self-targeting spacers. The number organisms with exactly 2 self-

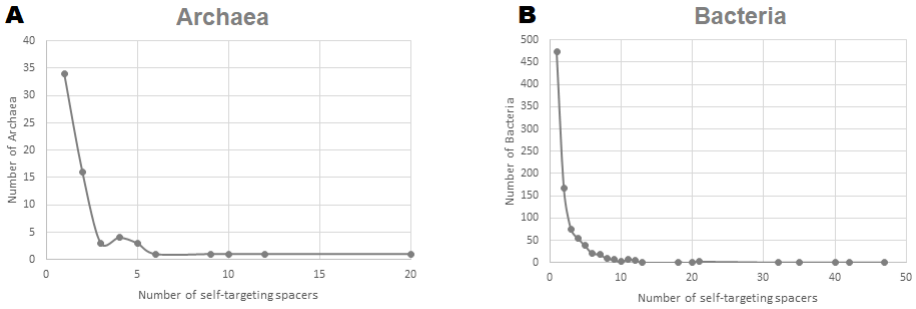

Figure 5.2: The distribution of self-targeting spacers in: (A) Archaea and (B) Bacteria.

targeting spacers was almost the same as those with 3 or more such spacers. Separately, only 5 of 163 (3%) self-targeting spacers were found on plasmids, and the remaining spacers being located on chromosomes.

5.4.3 The spread of self-targeting spacers in Bacteria

We found that 2325 self-targeting spacers were spread across 892 of 3058 (29.17%) bacterial organisms (Figure 5.2B). More than a half of the organisms with self-targeting spacers, 473 of 892 organisms (53.03%) had only one self-targeting spacer, and 167 organisms (18.72%) had exactly 2 self-targeting spacers. The remaining 252 organisms (28.25%) had from 3 to 47 spacers. Separately, we noted that only 32 of 2325 (1%) self-targeting spacers were located on plasmids, all the remaining spacers were found on chromosomes.

5.4.4 The average length of spacers in CRISPR arrays of Archaea and Bacteria

We found that Archaeal spacers are longer on average than Bacterial spacers (Figure 5.3). The difference between the means evaluated using a two-sample two-sided

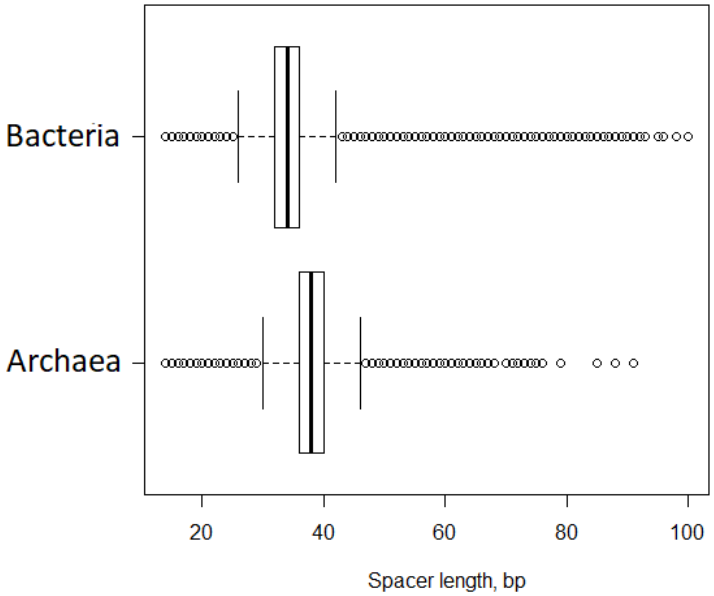

Figure 5.3: The distribution of spacer lengths in Archaea and Bacteria (spacers < 100 bp are shown).

t-test with unequal variance is statistically significant ($p < 2.2e-16$); the 95% confidence interval for the difference between the means is (3.54,3.64). Interestingly, self-targeting spacers in Archaea tend to be shorter than archaeal spacers overall and spacers without self-targeting (Table 5.4). Self-targeting spacers in Bacteria is about the same length as bacterial spacers overall and spacers without self-targeting. However, the standard deviation is high compared to the difference in the means, so the difference is likely not statistically significant.

Table 5.4: The average length of spacers in CRISPR arrays of Bacteria and Archaea.

Spacer length (mean ± sd)	Archaea	Bacteria
All spacers	38.22 ± 3.96	34.66 ± 5.22
Self-targeting spacers	33.83 ± 7.28	35.33 ± 8.57
Spacers without self-targeting	38.24 ± 3.92	34.64 ± 5.15

5.5 Discussion

The autoimmunity problems are a factor of evolutionary pressure on prokaryotes that possess CRISPR systems. Our findings demonstrate that about a third of prokaryotes carry self-targeting spacers even though the fraction of self-targeting spacers in a pool of all spacers is rather small ($\sim 1.5\%$). We found a significant difference in self-targeting rates between Bacteria and Archaea ($p < 2.2e-16$). Although Archaea on average possesses several times more spacers in their genome than Bacteria, the rate of self-targeting spacers in Archaea is almost three times lower than in Bacteria. This suggests that Archaea have developed more robust mechanisms of CRISPR systems and can manage larger spacer memory. Consequently, Archaea may accumulate more spacers and have a lower turn-over of spacers than Bacteria.

We also found that Archaea tend to have slightly longer spacer on average than Bacteria. It means archaeal spacers are more specific in capturing potential invaders. The longer spacer can also explain the decrease in the number of self-targeting events since higher spacer specificity protects better from spurious matches.

In addition, we found that self-targeting spacers in Archaea have shorter length in comparison to the average length of spacers overall. Thus, self-targeting events might be driven by acquiring spacers with not enough specificity. However, for Bacteria,

the problem of self-targeting might have a different origin since they have about the same average length for self-targeting and other spacers. Considering very intensive genomic exchange in Bacteria, the increased specificity might not be helpful because of extensive fragments shared between phages and Bacteria. In this case, self-targeting is an embedded cost of genome flexibility. Also, we found that only 1-3% of self-targeting spacers in prokaryotes are present on plasmids. The fitness cost of plasmids that bear self-targeting spacers is usually less that the cost of self-targeting spacers on chromosomes, and such plasmids are often eliminated from genomes.

5.6 Conclusion

The induction of autoimmunity during the operation of CRISPR-Cas systems represents a potential opportunity for the selective destruction of pathogenic microorganisms. Our finding that CRISPR systems in Bacteria are more prone to autoimmunity may provide important opportunities to develop new treatment methods that are alternative to antibiotics. Future studies may explore two possible directions: (a) how the pressure of autoimmunity shapes the evolution of bacteria and (b) how we can use autoimmunity manifestations for treatment of bacterial infections.

Chapter 6

Host-parasite associations

6.1 Abstract

A purely data-driven computational method is proposed to identify shared signals of genome interactions among sets of organisms from unannotated genetic sequence data. The method is based on extracting and manipulating long genomic strings and is scalable to long genome lengths. Unlike alignments, it can identify multiple copies of motifs, even on a mix of forward and reverse strands. It can achieve higher specificity compared to traditional k-mer methods. Also, it can be adapted to computing areas of putative transfer of genetic material, detect multiple copies of transferred subsequences, and can be used as a basis to analyze bacteria/viruses based on their functional interactions. The method can act as an initial screening tool to estimate the host range of prospective phage candidates for treatment of bacterial infections and other medical/biological applications. We illustrate the method using genetic sequence data on: (a) 2699 bacteria and 820 viruses from (Edwards et al., 2016) and (b) 3244 well-characterized prokaryote replicons (chromosomes and plasmids) and 1962 prokaryote viruses. This chapter is based on our report at the American Society for Virology (ASV) Annual Meeting (Lenskaia and Boley, 2019).

6.2 Background

Advances in sequencing and genome assembling technologies have led to the burst of newly discovered viruses without the need of culturing them in a lab using low-throughput wet-lab methods. Numerous new viral sequences have come as prophages from mining microbial genomes (Roux et al., 2015) and viral contigs from microbiome samples and metagenomic research (Manrique et al., 2016; Paez-Espino et al., 2016; Mokili et al., 2012). All this data leads to the development of new ways to obtain putative (a) taxonomy assignment and (b) host detection.

Taxonomy identification is generally determined by a canonical human-centered classification process under the auspices of the International Committee for the Taxonomy of Viruses (ICTV) using various criteria such as morphology, genome structure and segmentation, sequence similarity, etc. This process is too laborious to keep up with the mass of new genomic data continually produced and leads to a need for a purely computational screening approach, at least at a preliminary stage. One of the promising approaches is network-based classification based on gene-content similarity between viruses (Iranzo et al., 2017). This approach is more appropriate for viruses than the canonical tree-of-life structure since viruses have mosaic genome organization and high level of genomic exchange (Casjens, 2003). Moreover, there are no genes shared across all viruses suitable for their classification analogous to the 16S rRNA gene for prokaryotes.

Previous research (Bolduc et al., 2017) demonstrated that network-based and canonical authority-based taxonomy identification of viruses are usually in agreement as long as there are sufficient genomic similarities with known viruses but can disagree for viruses subject to extra high genomic exchange with several groups of viruses. A high genomic exchange rate between organisms that distorts traditional taxonomy identification of viruses might actually be useful in detecting new host-

parasite associations.

It is worth noting that the information about the found host-parasite associations for viruses is usually very limited. When mining prokaryote genome data for prophages, the prophages found are assigned to the associated host. Many databases (like NCBI) generally allow space for only one host, making it difficult to track situations when a given virus interacts with a wider prokaryote population. We seek a scalable method to automatically screen genomic sequence data for putative host-parasite associations while not leading to excessive false positives.

Many existing methods that can be utilize for detecting host-parasite associations by purely computational means were reviewed by (Edwards et al., 2016) including the longest exact match, alignments (Schuler et al., 1991; Altschul et al., 1997; Chen et al., 2015), CRISPR-Cas interactions (Barrangou and Van Der Oost, 2013) including CRISPR spacers identity and number of hits, co-abundance profiles (Stern et al., 2012), GC-content, and oligonucleotide frequencies (k=3-8 bp). The core methods for substring counting can be implemented very efficiently (Marçais and Kingsford, 2011; Melsted and Pritchard, 2011) with the further improvement using optimization schemes (Marçais et al., 2017). The accuracy of host predictions was accessed at different taxonomic levels (species, genus, family, order, class, and phylum). The accuracy was higher for the broader taxonomic units at a cost of lack of specificity in the predicted hosts. Among the reviewed methods (Edwards et al., 2016), the best performance at species level was reached by the longest exact match with an accuracy of 40.5% when accepting the top 4 candidates.

Also, alignment-free methods like longest exact match have been found to be most useful in certain applications involving whole genome sequence comparisons. Alignment-free methods can often avoid limitations of alignment-based methods including genome rearrangements (duplications, large insertions/deletions), horizontal gene transfer (HTG) events and highly divergent sequence comparisons (Ren et al.,

2018). Alignment-free methods can also be much more efficient and scalable when scanning large databases of sequences, but it is important to choose an appropriate word-length for the underlying application (Ren et al., 2018).

The analysis of (Edwards et al., 2016) has led to follow-up comparisons using purely computational analysis of genomic data. Ahlgren et al. (2017) used the same data to benchmark performance of different similarity scoring methods, both direct and adjusted for background substring frequencies, achieving an accuracy of 26% ($d_2^*, k = 6$) at the species level, the best among 11 compared similarity metrics including five background neutral measures for evaluating distances between vectors of string frequencies (Jensen-Shannon divergence (Narlikar et al., 2013), Chebyshev distance, Manhattan distance, Euclidean distance, and d_2 (Blaisdell, 1986)) and six background normalization methods (Willner (Karlin et al., 1997), Teeling (Teeling et al., 2004), EuF (Pride et al., 2006), Hao (Qi et al., 2004), d_2^* and d_2^s (Reinert et al., 2009)). Additional optimization steps were applied to improve performance including thresholding and computing consensus host to improve accuracy at higher taxonomic levels. Zhang et al. (2017) applied machine learning techniques to enhance the scalability of methods for detecting host-parasite associations for the large amount of accumulated data. All these other methods measured performance by assuming that the listed host-parasite pairs were positive examples and all other possible pairs were negative examples, but it is likely that the many negative examples might include undiscovered host-parasite interactions.

Although the longest exact match was one of the more successful criteria, it is limited to a single match per host-parasite pair. By recording many other possible long matches, we can recover many other putative host-parasite interactions. Hence we would like to use a scalable methods to record the existence of many long matches between putative host-parasite pairs. By tracking multiple matches of various lengths, we hope to be able to detect not only recent host-parasite associations, but also

interactions that might have occurred in the past and were partially obscured. We would like to compute a measure that would allow us to vary the specificity beyond Edward's somewhat arbitrary choice of keeping the top 4 contenders for host.

In this study, we propose a method based on tracking long sequences shared between hosts and phages. Instead of using frequencies of occurrence for short k-mers (Vinga and Almeida, 2003; Castellini et al., 2012), we use strings long enough so that the probability of being shared by two unrelated organisms is practically nil, but still capable of detecting many host-parasite associations. Using statistical simulation, we determined that the presence of shared strings of length 40 between a bacterial host and a putative phage was a suitable indicator of a biologically significant interaction, achieving a 43% accuracy on the data of (Edwards et al., 2016) at the species level (higher than the best method reported therein). We also validated the method on a more recent dataset of 1962 viruses and 3244 well-characterized prokaryote replicons including chromosomes and plasmids to show the performance on the extended set of viruses and a set of well-characterized prokaryote sequences. By using a dictionary (hash-table) implementation, we observed that the algorithms are very efficient and scalable.

The method can yield several putative biologically significant host-parasite pairs, helping to identify viruses that have a wider scope of interaction beyond a single host or a single genus. This could be critical in certain applications. For example, to evaluate a phage candidate for phage therapy applications (Ventola, 2015; Abedon et al., 2011), it is critical to assess its potential host range in addition to its lytic activity with respect to the target bacterial pathogen. A broad host range can significantly restrict phage biosafety in medical applications because of possible non-specific microbiome disruption. It is important to have a fast preliminary-screening computational technique for phage host range detection that can take advantage of existing databases of complete genomes in addition to the existing wet-lab techniques (Kutter, 2009).

6.3 Methods

6.3.1 Datasets

We used two datesets: (1) old benchmark to compare the performance of different methods from Edwards et al. (2015); (2) more recent larger dataset to show the effectiveness on the more recent data. The first dataset allowed us to evaluate the performance of our methods in comparison to other existing methods both reviewed by Edwards and developed afterwards (Ahlgren et al., 2017; Zhang et al., 2017). The dataset contains 2699 bacteria and 820 phages. The list of genomes was downloaded from Edward's website http://edwards.sdsu.edu/PhageHosts/. Genomes from the list were downloaded from NCBI Nucleotide database in gbk format. Each phage in the dataset had at least one annotated host mentioned in the host field of its annotation.

The second "bacteriophage" dataset consisted of 1962 bacteriophages downloaded from NCBI Virus database (Hatcher et al., 2017) on March 2019 of which 1765 phages have information about at least one annotated host on their records (host or lab_host fields were non-empty), plus 3422 well-characterized prokaryote genomes including both chromosomes and plasmids for complete reference and representative prokaryote genomes. For example, *E.coli* has 6 genomes that are denoted (labeled) "reference and representative", but one of these contains two sequenced plasmids and no chromosome sequenced. Hence only those 5 were included in the dataset. We evaluated the performance of our methods on the subset of viruses for which the host is known. We also make host prediction for viruses that do not have information about their host on file. Both lists, prokaryotic genomes and viral genomes are available online at https://github.com/tlensk/PhageScreen/.

6.3.2 Experimental design

Our proposed method is based on collecting all strings of a given length m that are long enough so that the probability two unrelated organisms sharing any such strings is essentially negligible. Our plan is to form dictionaries of all strings of a given length m present in the organisms of study. It is then a simple matter to find common strings between different organisms (indicating the high possibility of genetic interactions) or duplication of shared genetic sequences indicative of a history of multiple interactions over time. The resulting intersection matrix can then be used for a variety of analyses. Here we use the matrix of intersection ratios to identify putative host/parasite pairs of biological significance, to identify cases of multiple genetic transfers between individual phages and their hosts.

6.3.3 Choice of string window length

The choice of window length m is a critical parameter for the success of our analysis. We use statistical modeling to determine an appropriate string length m to reduce the probability of nonempty intersections due to random coincidence to a negligible level (specificity). We also check that our window lengths are not so long that we lose all intersections even between known host/phage pairs (sensitivity), and compare our window lengths to those used by bacteria in nature to defend against viral invaders via the CRISPR mechanism (Barrangou and Van Der Oost, 2013).

To estimate specificity, we first determine a null-hypothesis to evaluate the probability of obtaining a non-empty intersection between two randomly generated genomes. Since the strings of length m arise from a sliding window, they are not statistically independent, so they cannot be modelled by a simple statistical distribution such as a multinomial distribution. Hence, we use a numerical simulation described in Chapter 3.

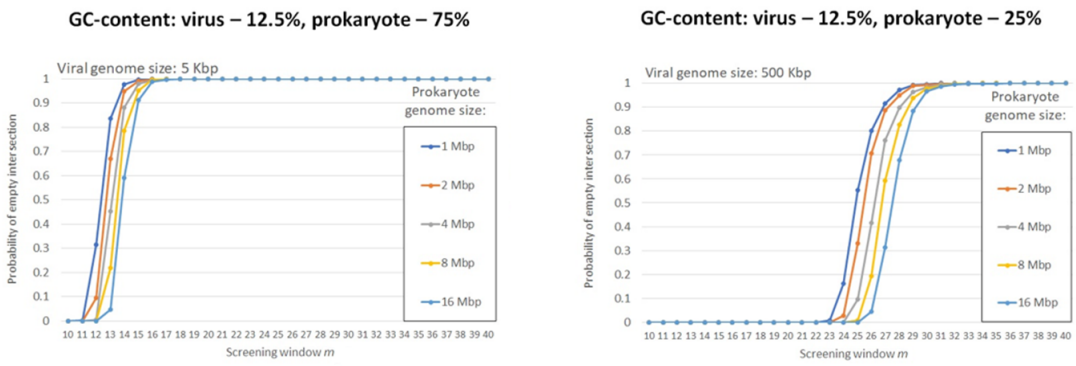

Figure 6.1: Simulation results for uniform distribution and two extreme skewed cases for GC-content combinations: (1) phage – 12.5%, bacterium – 75% (left) and (2) phage – 12.5%, bacterium – 25% (right). The vertical axis shows the fraction of randomly generated "host/parasite" pairs that had empty intersection 1024 trials ($p < .001$).

Figure 6.1 shows the extreme cases: for short window sizes ($m < 10$) every randomly generated pair had a non-empty intersection, while for $m > 32$ no pair had a non-empty intersection in 1024 trials. The extreme cases were reached with longer viruses sharing low GC-content with bacterial hosts and shorter viruses with GC-content opposite to that of the bacterial hosts. In all other cases, including balanced GC-content, the transition from all to none were in between these extreme cases.

Also, the results of our previous study about prokaryote autoimmunity potential (Lenskaia and Boley, 2020a) indicate that CRISPR-Cas systems in Archaea that are less prone to self-targeting use spacers that have average length close to 40 bp. These findings combained with the results of statistical modeling suggest that a window length of $m = 40$ would provide a good balance between sensitivity and specificity.

6.3.4 Location of fragments

Having information about the intersection between genomes, it is possible to map this intersection back to each of these genomes to obtain information about fragments'

location. Information about the location is useful for biological interpretation of findings (Sievers et al., 2017). This mapping allows us to identify occurrences of biological significance such as multiple copies of segments of viral genome present in bacterial genomes, and segments in both the forward and reverse strands. We made this mapping fast by incorporating information about frequencies and positions of each unique string to genome dictionaries. It allowed us to get information about fragment location in a linear time by going through the list of strings in the intersection and looking up the positions for each string in both genome dictionaries.

6.3.5 Detecting shuffled fragments

By recording the locations of the matched strings, we can quickly identify shuffled fragments, namely fragments that are copied from one genome to another but in a different order. For example, suppose we have four shared fragments in one genome ordered f1, f2, f3, f4. These fragments may appear in the second genome in the shuffled order f2, f4, f3, f1. The difference can result not only from the order of fragments but also from the copy number variation of certain fragments in the compared genomes (e.g., genome1: f1,f2,f3,f4 and genome2: f1,f2, f1,f3,f2, f4).

Alignment-based methods can capture shuffled fragments, but only if their lengths are long enough. If the shuffled fragments include a mix of long and medium length fragments, alignment methods could easily miss the shorter fragments. For example, *E.coli O104:H4 str. 2011C-3493* (NC_018658) and *Enterobacteria phage 933W* (NC_000924) share three long fragments (plus/plus comparison) we will denote f1, f2, f3 of length 2000-4000 bp plus a shorter fragment f4 of length 200 bp. In the bacterial genome these appear in order f1, f2, f3, f4 but in the phage they appear in the order f4, f1, f2, f3, as detected by our dictionary method. However, an alignment method focusing on the longer higher scoring alignments may discard the short fragment f4.

Our dictionary method can find shared fragments regardless of their relative order

in genomes. It is good for preliminary screening of compared genomes to get an estimate that can be further refined by alignment or additional screening using smaller window sizes.

We use exact matching to avoid any possible ambiguity, and we still find many significant matches in spite of the apparent "rigidity" of exact matches. Using alignments allows researchers to tolerate minor changes in related sequences at cost of introducing some degree of uncertainty that is hard to measure. Recently developed exhaustive alignment methods, e.g., BURST (Al-Ghalith and Knights, 2017) guarantees to find all possible alignment above a defined threshold but the main problem is that it generates many spurious alignments that need to be filtered out. We plan to relax the stringency of exact matching in the future. However, the number of existing exact matches allow us to take advantage of being certain about not only the hit itself but also about the location of this hit.

6.4 Results

To obtain results that we can evaluate, we compare our method to the results obtained by (Edwards et al., 2016) on the same data set, and then also apply the method to a more up-to-date dataset. The dataset used by Edwards et al. (2016) and later by other researchers (Ahlgren et al., 2017; Zhang et al., 2017) consists of 820 bacteriophages and 2699 bacterial gnomes, accordingly our method constructs two 2699x820 matrices of intersection ratios, one for forward strands and one for the reverse strand on the bacteria (forward on the virus). Since the lengths of the genomes and GC content in this dataset were similar to those assumed in the statistical modelling, a window length of $m = 40$ bp was found to be suitable. Using a window length of $m = 40$ bp, we found only 450 phages (54.9%) had a non-empty intersection with any bacteria in the dataset, while 370 phages had no intersection whatsoever. Also 1259 bacteria had

no intersection with any phages in the data set. In all, 14,387 bacterium-phage pairs were found with non-empty intersection, 0.33% of the total possible, counting both the forward and reverse strands. Of these 14,387 pairs with non-empty intersection, 38.64% had an intersection ratio of at least 0.01 on at least one strand (forward or reverse).

6.4.1 Identify putative host-phage pairs

In the Edwards et al. dataset, some species of bacteria were represented by many distinct strains (e.g., some species had more than 30 strains), while other species contained only one strain. Some strains had non-empty intersection with as many as 64 different phages, while other had non-empty intersection with only one. Figure 6.2 displays the number of phages with non-empty intersection with different strains of 5 bacteria with the most number of strains in the data set vs bacterial genome length. Strains of the same species and length had non-empty intersection with a similar number of phages, forming rather distinct clusters. Two outliers located far from their clusters are *E.coli str. K-12 substr. MDS42* and *S.aureus subsp. aureus VC40*. The former is a strain of *E.coli* with an artificially reduced genome (Pósfai et al., 2006). The latter is a laboratory derived strain of *S.aureus* that was engineered by a series of genome rearrangements and deletions to obtain resistance to antibiotics (Sass et al., 2012).

The larger number of phages overlapping with bacterial genomes generally correlates with pathogenicity of respective bacteria (Touchon et al., 2016). Two *E. coli* strains with the largest number of phages detected by our method include the pathogenic strains: *E.coli UMNK88* (point (a) on Figure 6.2) and emphE.coli O26:H11 str. 11368 (point (b) on Figure 6.2). However, the large number of detected phage inclusions may also reflect a history of the changes in biotechnological strains whose properties have been altered using viruses as vectors and which have had phage genes

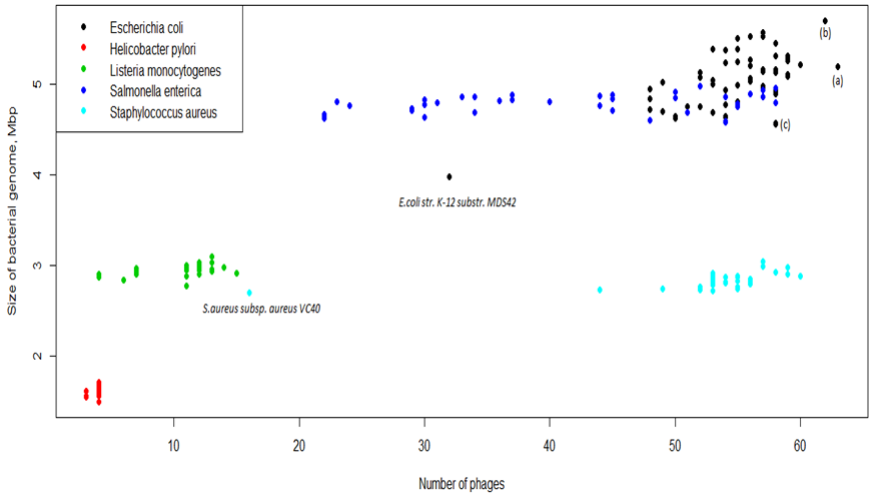

Figure 6.2: Number of phages identified in a host genome vs. size of bacterial genomes for 5 bacterial species that have large numbers of genomes for strains in the dataset.

inserted on purpose, e.g. a chemically competent strain of *E.coli BL21(DE3)* (point (c) on Figure 6.2).

In order to compare the performance of our method to those reported in (Edwards et al., 2016), we not only applied our method to the same data but also adopted the same methodology to accept a predicted host if it appeared among the list of potential hosts, i.e., the list of bacteria with nonempty intersection for each particular phage. We did not impose any additional threshold on the number of potential hosts found by our methods since the chosen window length was long and specific enough to avoid spurious matches. In our method, the matches were sorted by intersection ratio. Using this criterion, our method correctly predicted hosts for 43.8% phages (359 of 820 phages) in the dataset, compared to 40.5% for the best method reported in (Edwards et al., 2016) (Figure 6.3). Of these 359 phages with a "correctly predicted" host, the annotated host was found to have the highest ratio value among the bacteria in the dataset for 332 phages, while for the remaining 27 phages the annotated host appeared in the list of predicted hosts, but was not the bacterial strain with the largest intersection ratio. For example, *Burkholderia phage phiE125* (NC_003309)

had *Burkholderia thailandensis* as its annotated host. Although the intersection ratio for this host/phage pair was rather high (0.25), this phage was found to have a much larger intersection (0.34) with *Burkholderia pseudomallei*.

Our findings were consistent with the previously reported results (Edwards et al., 2016) for two *Burkholderia* phages, *Bcept176* and *KS5*. We found that these phages were incorporated as prophages to *Burkholderia multivorans ATCC 17616*, chromosome 2 (NC_010805). Both have a different bacterium of the same genus as its annotated host. Although *Burkholderia phage KS5* (NC_015265) has *Burkholderia cenocepacia* as its annotated host ($ratio = 0.0144$), it is entirely incorporated in *Burkholderia multivorans ATCC 17616*, chromosome 2 (NC_010805) ($ratio = 1$). *Burkholderia phage Bcep176* (NC_007497) has *Burkholderia cepacian* as its annotated host with a relatively small value of the ratio (0.0007). However, this phage was almost entirely incorporated ($ratio = 0.9994$) in another bacterium from the same genus, *Burkholderia multivorans ATCC 17616*, chromosome 2 (NC_010805). The annotation for *Burkholderia multivorans ATCC 17616* does not contain information about these events even though it has some predicted putative genes in these locations.

In addition, we found *Burkholderia phage KS5* (NC_015265) to be entirely incorporated in another sequence of *Burkholderia multivorans ATCC 17616*, chromosome 2 (NC_010086), but on the reverse strand. Our dictionary comparison of these two bacterial sequences of chromosome 2, NC_010805 and NC_010086, demonstrated very low sequence identity, below 0.5% ($m = 40$bp). However, the dictionary of the reverse complement of one sequence exactly matched the dictionary of the direct strand of the other sequence. These two entries seem to represent very closely related strains or even the same strain that were sequenced and assembled in the opposite directions.

Of the 820 bacteria, 450 had a non-empty intersection with some virus while 370 did not. Of the 450, 359 had significant non-empty intersection with their annotated

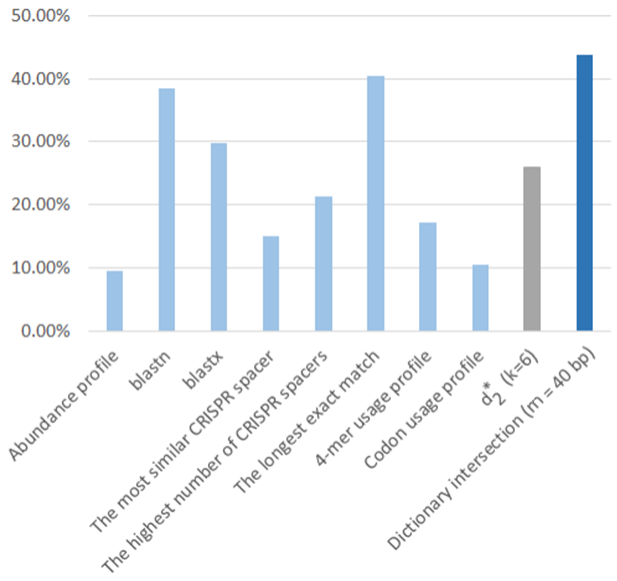

Figure 6.3: Percentage of correctly predicted hosts at the species level by different methods: (1) light blue bars represent previously obtained results (Edwards et al., 2016); (2) a grey bar represents the result reported by Ahlgren et al. (2017) on the previous dataset (Edwards et al., 2016) for the species level prediction; (3) dark blue bar is the result of our method applied to the same dataset. This comparison does not include accuracy of the methods developed by Zhang et al. (2017) due it was not reported.

host. For the remaining 91 phages, the annotated host was not in the list of bacteria with a non-empty intersection. For 40 of these 91 phages, the phage did have non-empty intersection with bacteria of the same genus as the annotated host. Among the remaining 51 of 91 phages, we found a few cases with a large intersection ($ratio > 0.3$) between a phage and a bacterium that was rather taxonomically far from the annotated host. For example, *Yersinia phage L-413C* (NC_004745) had $ratio = 0.34$ with several strains of *E.coli*. A detailed analysis (Garcia et al., 2008) of this case revealed that this phage is very similar to *Enterobacteria phage P2* (NC_001895). In addition, we also found large fragments matching *Yersinia phage* in other species of bacteria, e.g., *Salmonella enterica* and *Shigella sonnei*, but not in the twelve strains of *Yersinia pestis* that were present in the dataset. Although *Yersinia phage* has acquired genes ensuring its ability to infect *Y.pestis*, it has lost its ability to integrate itself into the genome of this new host. Another example is *Staphylococcus phage SpaA1* (NC_018277) that has *Staphylococcus pasteuri* as its annotated host with no intersection. However, this phage had a large intersection ($ratio = 0.42$) with *Bacillus thuringiensis serovar kurstaki str. HD73* (NC_020238). This phage genome represents an interesting case of a chimeric genome (Swanson et al., 2012).

6.4.2 Identify specificity of phage interactions

The matrix representation of the genomic intersections has allowed us to focus on different sections of host-parasite relationships. The matrix can help quickly identify bacterial strains that interact with a large number of phages vs those strains interreacting with only a few, and also distinguish between phages which interact with many bacterial strains versus those that interact with only one or two strains of a specific species. For example, *E.coli* had 62 genomes represented within the dataset. A total of 81 phages had a non-empty intersection at 40 bp with at least one of these *E.coli* strains, but only 26 of these phages had non-empty intersection with all *E.coli*

strains.

For instance, *E.coli K-12 substr. MG1655* (NC_000913) had non-empty intersection with 50 phages within the dataset, of which 24 phages had non-empty intersection with only some *E.coli* strains in the dataset (from 14 to 61 strains). Moreover, the amount of intersection varies from one strain to another and correlates with strain properties. Thus, the number of identified phages can indicate functional properties of bacterial hosts. Pathogenic stains and biotechnological strains tend to have a larger number of incorporated phage fragments. Thus, it allows researchers to distinguish between different strains based on the amount of their intersection with phages (Lenskaia and Boley, 2018).

6.4.3 Identify phage-host transfer in genes and intergenetic regions

So far, we have used the string matching algorithm only to identify host/parasite pairs with significant intersection among the genomes. While constructing the individual genome dictionaries, we can record the locations of the individual strings within the respective genome and use this to find the locations of the shared fragments in the bacterial and phage genomes, including whether it is to be found on the direct strand, reverse strand or both. Among 26 phages that have non-empty intersection with all *E.coli* strains in the dataset, we found 8 phages (available in the supplementary materials) for which the bacterial genome of *E.coli K-12* contains multiple copies of some phage-genome fragment matching these phages. We found one shared fragment that appears twice in the bacterial genome and once in each of the 8 phages. For the two fragment copies we found, one was within the known tRNA Thr gene (position 262901..262948) and another one was in an unannotated intergenic region (position 297209..297256).

We also found a phage containing multiple copies of the fragment that appeared only once in the bacterial genome. The order of shared fragments in bacterial and phage genomes were often shuffled, a situation that would be hard to capture by alignment methods. In addition, we found 24 cases when the entire phage was inserted into bacteria ($ratio = 1.0$). In most cases, the phage genome was present in the bacterial genome as one contiguous segment, but in all but 4 cases, copies of smaller fragments were also present elsewhere in the bacterial genome, sometimes on the reverse strand too. Interestingly, the intersection in these cases besides this entire prophage fragment often included many smaller fragments on both forward and reverse strands in bacterial genome that matched the same phage.

6.4.4 Bacteriophage dataset

For the new Bacteriophage dataset downloaded in 2019 consisting of 3244 replicons and 1962 viruses, we found 5106 prokaryote-virus pair with overlapping genomes (¡0.1%). In this dataset 197 viruses do not have the annotated host information on file. For 71 of 197 viruses, we found overlaps with some prokaryotes. This might help us infer possible host range for these viruses with unknown host. Following the protocol used for the Edwards et al. data set, we considered for further analysis only viruses that have the annotated host description at least at the species level and at least one sequenced genome of their annotated host in the dataset. This eliminated 384 viruses in addition to viruses without an annotated host. Thus, the final set of viruses for host prediction analysis contained 1381 viruses. Table 6.1 summarizes the results of this analysis.

Our method provides an opportunity to analyze forward and reverse strands separately. Unlike previous work based on k-mers, we distinguish between fragments found on the forward strand and those found on the reverse strand. This allows us to discover differences of possible biological significance between the two strands.

Table 6.1: The results of the host prediction analysis.

Virus Category	Ed. DB	New DB
1. No host listed in DB and no intersection with any prokaryote host	0	126
2. No host listed in DB but has intersections with some host in DB	0	71
3. No sequence data for listed host	0	384
Total with no data (not part of the evaluation)	0	581
4. No intersection with any host in DB at $m = 40$ bp	370	880
5. Intersection with some prokaryote but not with listed host	51	66
6. Intersection with host of the same genus as listed host	40	75
Total disagreements	461	1021
7. Intersection with listed host but larger intersect with another host	27	43
8. Largest intersection is with listed annotated host	332	317
Total agreements	359	360

Figure 6.4 shows the intersection ratio for forward and reverse strands for all 5106 host-phage pairs that have a non-empty intersection. Among these pairs, 893 (17.5%) had overlaps on both forward and reverse strands and the rest were almost equally divided into only forward (2106) and only reverse (2107) groups. The top 3 pairs with overlaps on forward and reverse strands are parasitic bacteria from *Spiroplasma* genus and their viruses. The "dot plot" in Figure 6.5 shows there are multiple copies of the virus in the host genome some in the forward strand and some in reverse.

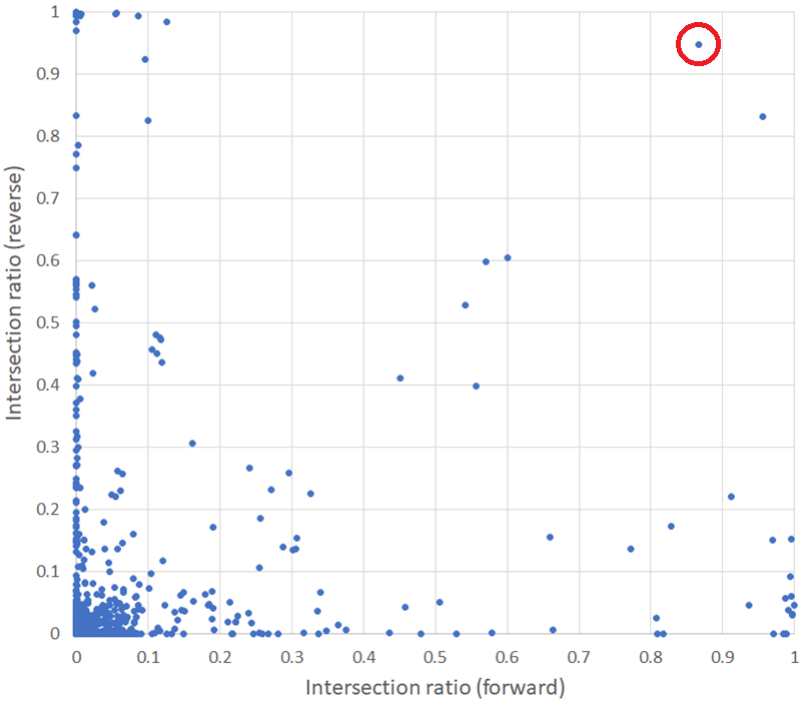

Figure 6.4: Forward and reverse strand intersection ratios for host-phage pairs with non-empty intersection. Details of matching intersection shown in Figure 6.5.

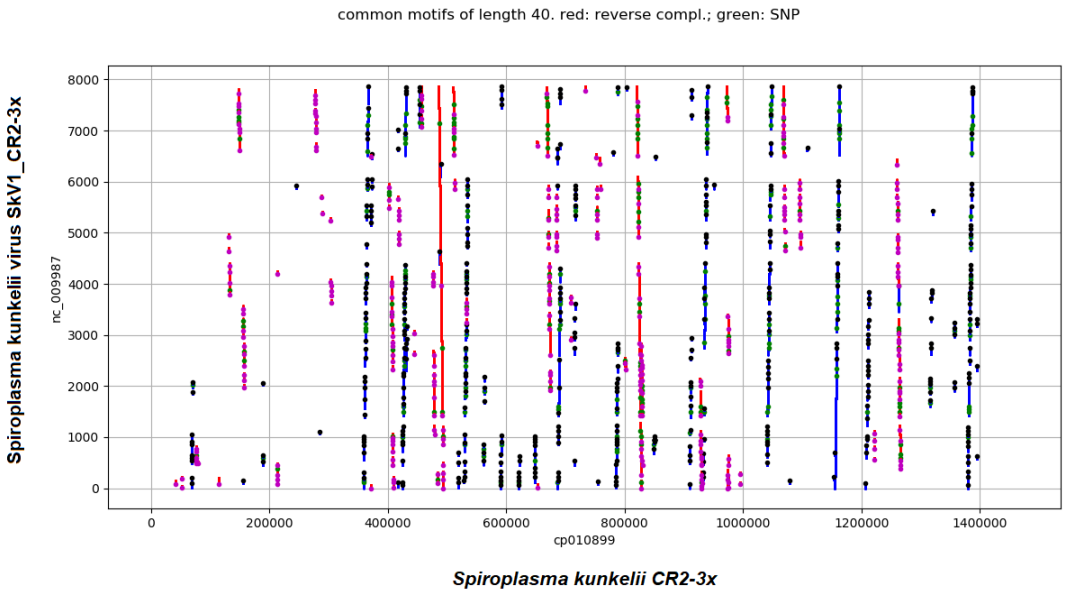

Figure 6.5: Intersection between NC_009987 virus and CP010899 host (the circled pair in Figure 6.4) plotted using the dot-matrix display. Red are matches to the reverse complement. Green dots are SNPs.

6.5 Discussion

Previously a virus discovery was closely linked with its host depiction. However, considering dramatical reduction of sequencing costs and advances in assembly technologies, many viral discoveries now are routinely made from microbiome sequencing and mining microbial genome without the need of culturing a new virus in a lab and identifying its host firsthand. It demands computational methods capable to extract signals of potential host-parasite associations primarily from genomic sequences. Although many methods to address this question are proposed, the question of their fair benchmarking remains open. Usually, prediction performance is evaluated based on the ability to guess the annotated host information that some viruses might have on file. However, the utility and completeness of this information are questionable. Information about one annotated host does not adequately described possible host-

parasite interactions. The existence of polyvalent viruses (Pantůček et al., 1998) challenges this approach to host description.

In addition, no information does not necessarily mean no interactions, they might not be tested. Moreover, there is no record of tested negative associations. However, researchers use this limited annotated host information to perform ROC-AUC (Edwards et al., 2016) analysis expanding it beyond the hit-miss scenario. In this case, the question of interpreting false positives and false negatives is challenging.

Moreover, the method performance can be significantly impacted by the structure of the dataset, i.e., the existing relationship between bacteria and viruses. We applied our method on two datasets and found a significant change in accuracy: (1) 43% in Edwards's dataset; (2) 26% for the updated dataset. We expanded the set of viruses but took a different approach in picking prokaryote genomes. We considered only reference and representative prokaryote genomes. It gave us a more balanced coverage of species, e.g., the update dataset contains only 5 *E.coli* reference and representative genomes comparing to Edwards'dataset with 62 genomes of *E.coli* strains. In addition, we tested our methods on the combination of Edwards's bacteria and the most recent update of viruses (August, 2019) in NCBI Virus with 2,631 complete RefSeq phage genomes. We followed the protocol developed by Edwards et al. (2015) to consider for the further analysis and prediction only viruses that had the known annotated host (at least at the species level) and which annotated host had at least one sequenced genome in the dataset. The reduced set of viruses after the filtering contained 1662 phages. The accuracy on this set of viruses and Edward's et al. set of bacteria was "averaged" at 37%. Thus, varying the scope and coverage of different prokaryote and viral species results in rather different values for accuracy, the measure of performance usually reported for predicting host-parasite associations (Edwards et al., 2016; Ahlgren et al., 2017), i.e., the number of viruses with the correct prediction relative to the total number of viruses. The use of alternative ways

of measuring performance in case of the annotated host is challenging since the validation of "false positives" and "false negatives" is questionable. Although the value of accuracy alone is not very informative to judge method performance for detecting host-parasite associations, computational methods that are capable to provide a big picture of genomic interaction between the sets of organisms are very important. Moreover, developing more adequate benchmarking technology is critical for methods comparison.

Our method aims to provide a big picture of interactions between genomes. It allows us to adjust the level of sensitivity and specificity to obtain a more detailed picture with the necessary resolution by varying screening window size. It also makes it possible to compare general patterns of genomic intersections, helps filter out "random" intersections, and prioritize found genomic interactions that are of interest for further detailed analysis.

The dictionary approach is a generalized method with respect to other string methods. It can be adapted to provide results that have been obtained from other string methods such as the largest common string method and oligonucleotide frequencies. The dictionary method implements a new paradigm in complete genome screening since it allows researchers to search in a "not to miss anything" paradigm, unlike many existing methods that aim to find only some very specific entities in genomes. Using the dictionary approach, we found evidence of multiple interactions between phages and bacteria. We detected a significant number of phages that had fragments inserted in bacterial genomes (Figure 6.2) that exceed the number of prophage insertions that are usually reported (Brüssow et al., 2004; Touchon et al., 2016). Moreover, this representation can help researchers to estimate the potential impact of phages on various properties of bacterial hosts (e.g., pathogenicity, virulence, and biofilm formation) using the knowledge about life traits of these bacteria. The obtained estimates of the number of potential interactions raise important questions about the

real number of existing interactions between phages and bacteria based on different mechanisms. However, further study and biological evaluation are needed for identified phages with significant genome interactions with many bacteria from different genera.

Although this approach allows us to accurately identify the integration of phages into host genomes, it has a limited ability to detect any interaction that does not depend on such integration. Alternative methods (e.g., plaque assays, co-abundance profiles) can be used to identify the hosts for purely lytic phages.

It worth noting that our dictionary method can outline the scope of potential interactions. It can capture both direct (i.e., host-parasite associations) and indirect (e.g., horizontal gene transfer) genome interactions. The next step will be to create methods for identifying and studying different types of intersections, filtering direct and indirect interactions, and searching for distinctive features that are inherent in each type of interaction. Further research is needed to detect and evaluate these features. It would also be interesting to incorporate information from different levels of resolution (e.g., 30-50bp). These questions need collaboration with biologists for further exploration and validation.

Thus, the proposed computational screening method can identify host-parasite associations between bacteria and phages more accurately than previous methods. It avoids limitations of those methods in estimating a potential host range. In addition, we developed statistical modeling to assess the sensitivity and specificity of our method and adjusted it for practical applications. The dictionary method will provide an important advantage over wet-lab methods as a preliminary screening tool for biological and medical applications, e.g., filtering candidates for phage therapy by eliminating phages with fragments that are often present in genomes of pathogenic strains of target bacteria.

In addition, phages contribute to life traits of bacteria (Touchon et al., 2016) and

can alter functional properties of their hosts (Brüssow et al., 2004). The developed methods allow us to quickly estimate the amount of phage contribution to bacterial genomes by screening for the intersection between the genomes. The results of our previous study (Lenskaia and Boley, 2018) showed that the obtained data about the genomic intersections were useful as the input data for machine learning algorithms to predict functional properties of bacteria based on their interactions with phages. The data on pairwise values of the intersection ratio between phages and bacteria could help to identify the impact of phages on pathogenicity of bacteria. We found a group of phages that were good indicators of pathogenicity in *E.coli* strains. These "indicator" phages can help diagnose pathogenicity in newly sequenced strains based on their intersection with the phages. The developed methods might be useful to explore the impact of phages on different properties of bacteria including virulence, antibiotic resistance, and biofilm formation.

6.6 Conclusion

In conclusion, our dictionary method is useful to capture and analyze genomic interactions between organisms in large-scale screening research. Since it can quickly capture many different interactions that were previously obtained using various methods separately, our method is beneficial for integrating biological knowledge into the big picture to validate and explore genome interactions between organisms on a broader scale.

Chapter 7

Computational approach to predicting epidemics

7.1 Abstract

Viruses need time to adapt their genomes to a new host. It is a gradual process that can be monitored using genome sequencing. We develop computational methods that provide opportunities to track genome changes and identify possible sources of genome exchange during viral adaptation based on the scalable computational analysis of available big genomic data. We applied our methods to 3168 Coronaviridae viruses stored in NCBI Virus database to find the prerequisites of the ongoing COVID-19 pandemic. The results of our research can contribute to predicting and monitoring of the future pandemics. This chapter is based on our report at the Institute for Molecular Virology (Lenskaia and Boley, 2020b)

7.2 Background

Before the COVID-19 pandemic, many people were convinced that the times of global pandemics had been a thing of the past and modern technologies could combat any known pathogens with time and enough effort. The World Health Organization devel-

oped strategic plans to guide international efforts in eliminating future epidemics, e.g. Global Strategy to Eliminate Yellow fever Epidemics (WHO, 2018). Unfortunately, this false feeling of security made it impossible to correctly assess the situation and develop an adequate action plan at the early stages of the COVID-19 pandemic, i.e., the beginning of the pandemic was perceived by many people as a bolt from the blue.

However, a more correct analogy for the current pandemic is an avalanche when the sudden descent of the avalanche is preceded by many events: the accumulation of snow, the lack of anti-avalanche measures, an unwary skier who cuts the slope, etc. Moreover, an avalanche in one valley can cause a chain reaction for avalanches coming down in other valleys. And the pandemic avalanche seems to be only gaining strength and is still far from stopping.

However, deadly viruses and particularly pathogenic bacteria do not appear out of blue. It takes time for them to adapt to new hosts and find vulnerabilities in the host defense mechanisms. Also, pathogens need time to rearrange its genome. This time should not be wasted. This gives us an opportunity to compute the emerging virus while the virus computes us.

Fortunately, now we have an opportunity to sequence genomes of pathogens, including especially dangerous pathogens, work with which in a laboratory requires advanced security measures. Also, it is possible to sequence the pathogens that had caused global health problems in the past to better understand their evolutionary trajectories. These large amounts data require new computational methods for their analysis.

The accumulation of big genomic data has lead to the development of powerful methods of computational virology. Also, the idea to use the available data to monitor and predict future viral pathogens gave raise to global initiatives such as the Global Virome Project (Carroll et al., 2018). However, many existing methods aim for analyzing changes in a specific location in a genome or in relation to particular

genes. Unfortunately, such methods are missing the full picture of genome changes that reflects the existing relationships and interactions between viruses. The observed partial pictures can often lead to different conclusions. Since the early days of pandemic there were numerous attempts to identify a possible origin of a new virus and detect animal reservoirs for its evolution before the successful transition to humans. The use of various methods for partial genome analysis lead different research groups to detecting different suspects among animal hosts including bats (Zhou et al., 2020), pangolins (Lam et al., 2020; Xiao et al., 2020), and snakes(Ji et al., 2020). Later snakes were exonerated (Anderson, 2020; Robertson, 2020).

For tracking evolutionary trajectories of emerging pathogens, researchers often do a phylogenetic reconstruction based on the comparison of genetic sequences. Such reconstructions are based on the molecular clock assumptions.

Unfortunately, Coronaviruses have very high recombination rate compared to other recently emerged viruses such as Ebola and Zika viruses. According to (Boni et al., 2020), "different parts of genome have different histories." Thus, the attempts to capture key points of viral evolution on its way to successfully infect humans using phylogenetic reconstructions have been challenging. Recombination can significantly affect the evolutionary rate estimation since it violates the basic assumption of phylogenetic reconstructions about the existence of a single phylogenetic tree. Also, the molecular clock is affected by the population size and other factors. To overcome the limitations, Boni et al. focused on analyzing ancestry in non-recombinant regions of coronavirus genomes. This approach allowed researchers to eliminate negative effects of high recombination rates on the phylogenetic reconstruction.

However, recombination events themselves contain important signals about the previous viral interactions and should not be discarded during the analysis. Rapid monitoring of global rearrangements in the genome requires efficient computational methods. The current focus of many methods on the search for changes in individual

genes may capture some of the consequences of such rearrangements. However, the existing methods are not aimed to evaluating the overall state of the genome. Understanding the prerequisites of viral transmission to new species is a challenge in viral research. We decided to explore the host range of Coronaviruses and the existing relationships of host with Coronaviruses from multiple genera. In our opinion, hosts that interact with Coronaviruses from multiple genera can be a suitable reservoir for viral adaptation on its way to infecting new species.

We seek for methods that can capture signals both from similarities due to common ancestry and due recombination events. The methods should be capable of evaluating the state of the entire genome since , in our opinion, transmission to a new host requires significant changes and adjustments in viral genomes. Thus, we decided to utilize the integral evaluation of genomic interactions using a similarity measure based on computing the number of shared fragments between genomes. To eliminate any additional sources of uncertainty in our analysis we focused on exact matching rather than alignment for estimating the similarity between genomes. The shared fragments for the similarity estimation should be long enough to avoid spurious overlaps and short enough to capture as much signal about viral interactions as possible. Thus, the optimal choice of a window size for exact matching is essential for careful estimations.

We created a similarity matrix for a set of 3168 Coronaviruses to analyze their interactions. Also, we applied linear algebra methods to find densely connected components in this similarity matrix and trace viruses that share fragments with several such components. In this regard, the minimum degree ordering algorithm (George and Liu, 1989) is quite unique since it allowed us to rearrange the similarity matrix and identify these densely connected viral clusters. Traditional clustering algorithms such as k-means clustering cannot take into account the density of interactions between viruses while separating them into clusters, and the clustering results depend on the initialization (i.e., the number of clusters, the initial location of cluster centers).

Our research is aimed to developing screening methods to quickly assess the state of the genome as a whole and its disposition in relation to the genomes of other viruses. The main purpose of this research is to develop a computational approach that allows researchers to analyze transmission potential of viruses and monitor critical changes in viral genome before it will cause a pandemic. Also, it will help address the following questions: (1) What core components do viruses that infect the same host have? and (2) What viral genome properties can be indicative of possible viral transmission to a new host? Knowing the current state and direction of the changes for genomes of emerging pathogens will give us time to prepare counter-measures to prevent a downing pandemic or stop it early.

7.3 Methods

7.3.1 Coronavirus dataset

We accessed NCBI Virus on April 9, 2020. This date represented a checkpoint for the amount of data publicly available for Coronaviruses at the very beginning of the pandemic. Our search for Coronavirus genomes with the status equals "complete" yielded 3168 sequences. We downloaded the complete genomes for these Coronaviridae viruses from NCBI for further analysis. The list of accession numbers and the code for the core methods are available on the GitHub repository: https://github.com/tlensk/VirusMonitor.

7.3.2 Similarity matrix

We applied a similarity measure as a diagnostic test to monitor possible sources that contribute to viral evolution and adaptation of COVID-19 to humans. We computed a similarity square matrix using a pairwise intersection (overlap) between genomes.

To calculate the amount of overlap, we created a dictionary representation for each genome by using the sliding window of length 40. This length was chosen based on the results of our previous study (Lenskaia and Boley, 2020a). Previousely, we found that the most advanced CRISPR-Cas systems in Archaea which are less prone to self-targeting than Bacteria had the average length of spacer very close to 40 bp. Also, more than 93% of spacers in known CRISPR-Cas systems stored in CRISPRCasdb (Pourcel et al., 2020) have length below 40 bp. Since such length is suitable for CRISPR-Cas systems and it allows prokaryotes to avoid spurious overlaps in most cases then, in our opinion, it should also work for discriminating spurious overlaps between viral genomes. We stored this genome representation as a hash table with strings as keys and string frequencies as the corresponding values. The number of strings shared between keys of two dictionaries reflected the overlap between genomes. We did not adjust this computation by string frequency since most of strings of length 40 in viral genomes were unique (appeared in a genome only once). The diagonal of the matrix reflected the size of genome dictionary for each virus (i.e., the number of keys). The length of the window was chosen short enough such that it still captures shared fragment between genomes but at the same time it is long enough to discriminate possible spurious overlaps between genomes.

7.3.3 Computational analysis

The suggested approach allow researchers to facilitate the high-throughput computational analysis of complete genomes and divide the computational analysis into three stages. The first stage of the analysis aims to compute the similarity between viruses based on the number of shared fragments between their genomes. The second stage aims to reveal the relationships between viruses based on the observed degree of similarity and find clusters of densely connected viruses. To do it, we apply linear algebra methods to sparse similarity matrix. The third stage of the analysis aims to analyze

host-virus relationships within the identified clusters of viruses.

7.3.4 Linear algebra methods

We applied the minimum degree ordering algorithm implemented in MathWorks MATLAB R2020b as symamd function (Larimore and Davis, 2020) to find highly connected components in a symmetrical sparse similarity matrix of 3168 genomes of Coronaviruses. The goal of our analysis was not to merely separate viruses into clusters based on their similarity but to find the most densely connected clusters of viruses. This task would be difficult to complete without the use of linear algebra methods. The minimum degree algorithm originates from the graph theory, and it aims to re-arrange the matrix based on the number of connections for each element in such a way that it brings non-zero elements closer to the diagonal and clamps zero elements in large blocks off the diagonal. Our application of this algorithm for clustering is unique. In linear algebra, the algorithm is usually used for matrix re-arrangement before applying the Cholesky decomposition. However, this algorithm worked effectively on sparse matrices for the purpose of our research and produced stable clustering results. The traditional clustering algorithms such as k-means clustering do not guarantee optimal cluster assignments, require to set some parameters such as the number of clusters, and the results of clustering may vary depending on the initialization. But the sub-matrix that correspond to each cluster can be rather dense. Thus the minimum degree algorithms is not suitable for the individual cluster analysis unless we can increase the sparsity by introducing the threshold and zeroing all entries below it. In this case, the results of the analysis rely on a choice of a threshold and many subtle similarities might be missed. However, there are traditional methods that work with dense matrices, i.e., cluster analysis and network analysis.

7.4 Results

We analyzed the information about hosts of Coronaviruses available at the beginning of the COVID-19 pandemic to see if it was possible to anticipate the emergence of viral threats before this pandemic occurred. We also analyzed hosts with a broad spectrum of Coronaviruses from different genera that usually are in a close contact with humans. Having permanent contacts with such hosts can increase a risk of spill-overs from animals to humans and ultimately increase chances of successful transmission to humans. The genome similarity between viruses can be an indicator of the existing relationships between viruses. In this work, we analyzed the mutual state of proximity for the complete genomes of Coronaviruses using a similarity matrix computed based on the total number of long shared fragments. This measure helped us evaluate the amount of biologically significant interactions.

7.4.1 Hosts analysis for Coronaviruses from different genera

Our goal was to explore factors that can contribute to the probability of viral transmission. We analyzed the host range for Coronaviruses in the set (Figure 7.3) and the scope of possible interactions between their hosts and humans. The host that has many Coronaviruses from different genera and has close interactions with humans might be a potential source of emerging viruses that might successfully infect humans. Among 3168 viruses, there are 28.98% Alphacoronaviruses, 52.34% Betacoronaviruses, 4.17% Deltacoronaviruses, 12.34% Gammacoronaviruses, and 2.18% unspecified coronaviruses.The total list of hosts included 101 entries captured at various taxonomic levels from the species name to the order level. Each host belonged to one of the two classes, i.e., either Aves (birds) or Mammalia (mammals). We observed the existing tight connection between viral genus and host class: Alphacoronaviruses and Betacoronaviruses infect mammals; Deltacoronaviruses and Gammacoronaviruses

usually infect birds. However, we found two exceptions: (1) there are Deltacoronaviruses that infect pigs and (2) Gammacoronaviruses that infect marine mammals (e.g., beluga whales). We divided hosts into 23 groups based on their taxonomic ranks. The majority of host groups (almost 74%) represented a family (e.g., Bovidae and Canidae) with the following exceptions: (1) in mammals, 2 host groups were formed at the order level, Chiroptera (bats) and Pholidota (pangolins); (2) one host group were formed at the suborder level Feliformia (cat-like mammals) including the families Felidae and Viverridae; (3) we considered humans as a separate host group apart from the other representatives of Hominidae,e.g., *Pan troglodytes verus* (a western chimpanzee) that formed a separate host group; (4) for birds, two host groups were formed at the order to combine several families that belonged to the same order.

Among the 23 host groups, 14 (61%) had Coronaviruses from only one genus, 8 (35%) can be infected by Coronaviruses from two genera (Alphacoronaviruses and Betacoronaviruses), and only 1 host group (4%) Suidae that includes wild boar and domestic pig is susceptible to Coronaviruses from three of four viral genera (Alphacoronaviruses, Betacoronaviruses, and Deltacoronaviruses). This makes pigs a unique animal reservoir that might facilitate the emergence of new viruses.

We analyzed the number of Coronaviruses sequenced for each host in the dataset. This number varies for different host groups from hundreds to just several sequenced viruses. We visualized the distribution of Coronaviruses from different genera for the host groups that had more than 10 Coronaviruses and included Pholidota even though this group had only 5 sequences (Figure 7.1). Of course, the larger number of the sequenced viruses does not necessarily correspond to the higher epidemiological pressure on these hosts. However, host groups that have many Coronaviruses from different genera should be monitored more closely. Also, the increase in the number of the shared fragments between viruses that infect different hosts might be one of the indicators of the possible shift in viral transmission. We considered the first sequenced

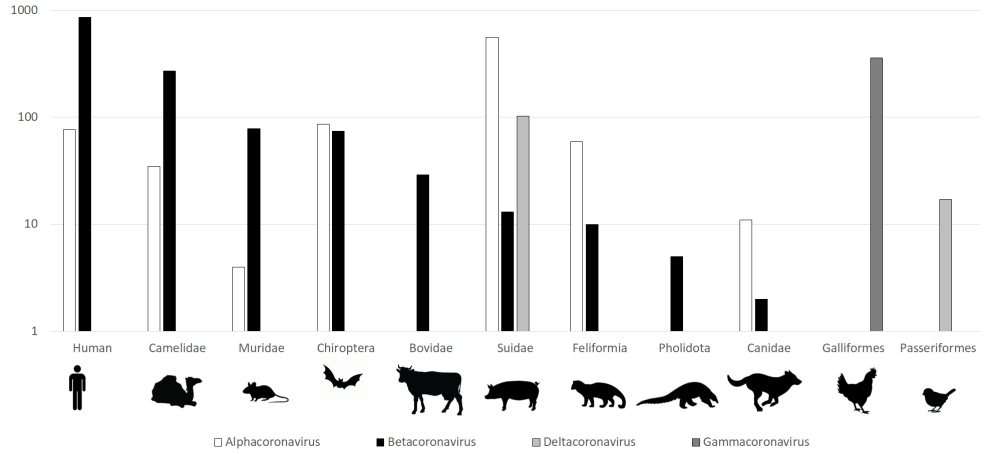

Figure 7.1: The number of Coronaviruses sequenced for each host in the dataset.

SARS-Cov-2 genome (MN908947) reported by researchers (Wu et al., 2020) as a point of reference for evaluating the transmission potential to humans.

7.4.2 Interactions between viruses from different genera

We analyzed the shared fragments found between viruses that belong to different genera. Although recombination events are very frequent for Coronaviruses within the same genus, there are only a limited number of fragments shared between viruses that belong to different genera (Figure 7.2). We found 2177 such pairs of viruses (0.02% cells in the similarity matrix). More detailed analysis showed that one of the Camel Betacoronavirus (KT368891) was previously incorrectly annotated as Alphacoronavirus. The exclusion of the pairs that contained this coronavirus left a set of 1880 pairs in which viruses belong to different genera and share at least 1 string of length 40. Also, we did not find any significant overlaps between the following viral pair combinations: (a) Alphacoronaviruses and Deltacoronaviruses and (b) Betacoronaviruses and Deltacoronaviruses.

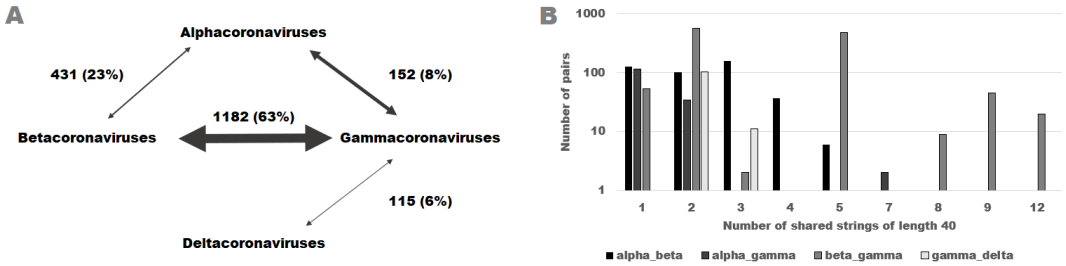

Figure 7.2: (A) The number of the identified pairs for different combinations of genera.(B) Distribution of shared strings of length 40 for the pairs in which viruses are from different genera.

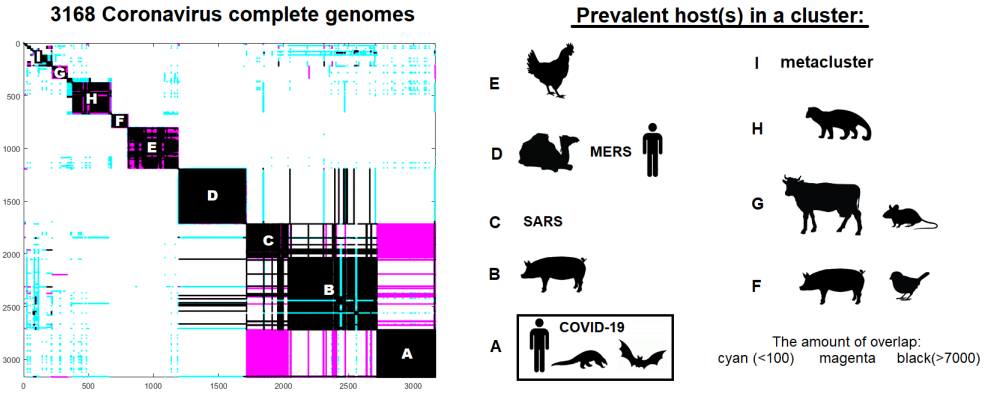

Figure 7.3: Blocks found in the similarity matrix of 3168 Coronaviruses after applying a minimum degree algorithm.

7.4.3 Similarity matrix analysis

Analysis of the similarity matrix between Coronaviruses using linear algebra methods showed the presence of several strongly connected components and highlighted ways of genetic exchange between them (Figure 7.3).

The clusters within the matrix reflect the groups of viruses most closely related to each other, as well as the existing connections between the blocks (Figure 7.3). We found 126 viral clusters. Only four of them were isolated clusters, i.e., viruses have some similarities within the cluster but have no shared fragments with any other

viruses outside their cluster. The remaining 122 clusters were connected (have some degree of connectivity) by having viruses that share fragments with several clusters. Among 122 connected clusters, there are 33 singleton clusters. These clusters consist of a single virus that act as a hub for connecting other clusters. Some clusters are organized in a way that they form a metacluster that contains viruses form several dense subgroups that are less densely connected between each other. A more detailed analysis of viruses within each of the blocks showed the existence of the dominant host(s) within the clusters.

We found two main types of clusters with multiple viruses: solid clusters and disperse clusters (Figure 7.3). Solid clusters contain very closely related viruses with very high level of similarity between their genomes, e.g., cluster A and cluster D. These clusters reflect the most recent interactions between viruses. Disperse clusters have multiple cores of viruses that are much densely connected with each other than with the rest of viruses in the cluster, e.g., cluster B, cluster C, and cluster E. Disperse clusters reflect more distant interactions when connections between cores in the disperse clusters are more relaxed. COVID-19 strains form a solid cluster, cluster A, in the bottom right corner of the matrix after applying the minimum degree algorithm. This cluster also includes several pangolin viruses and bat virus that were associated with the origin of the pandemic in animal species. Cluster B is an example of the disperse cluster with multiple cores. We found that this cluster contains many swine Coronaviruses that form a background for establishing connections between the core groups of viruses. Swines seem to play a significant role in facilitating evolution of strains that can overcome the transmission barriers between different species. It is possible that some challenges in viral evolution might be solved by considering the evolution of viromes and viral communities that include many different viruses rather than by focusing on the sole evolution of a given virus and its fellow viruses from the same taxonomy unit.

The found matrix representation allows to capture and analyze relationships between viruses at a large scale. This representation is very useful for identifying computational reservoir for viral evolution and key hubs of genome exchange. Figure 7.3 indicates that viruses are connected with each other in a way that looks like a network for distributed computing. Some clues and vulnerabilities in a host defense computed by some viruses can be extensively shared in the network. Moreover, viruses can adjust their genomes in a host using recombination. They are mainly intact to changes while they travel inside their cuspids as viral particles. Thus, to facilitate massive computations and the subsequent genome changes, it is critical to have computational platform such as a large epidemics in animals or in humans. We assume that metaclusters demonstrate the results of such massive computational efforts.

Any pandemic is preceded by local epidemics to get vulnerabilities in new host defense and some large epidemics in related hosts to combine, test, and distribute the obtained computational results to justify the necessary genome changes. In case of Coronavirus pandemic, the preceding local epidemics such as SARS and MERS indicated viral efforts to find keys to human immunity. Moreover, the preceding pandemics in animals can provide unprecedented computational resources to viral evolution.

7.5 Discussion

Our analysis of the similarity matrix indicated the existing attempts to unlock human defense by different groups of Coronaviruses. Some of these attempts were more successful than others including local epidemics of SARS (2002-2003) and MERS (2012). Those epidemics helped viruses find some useful clues to overcome human defenses. Also, the significant acceleration of viral evolution on a large scale can be achieved by the pandemic of related viruses in other species that are in close contact with humans.

Pigs might be one of such examples. According to our analysis, Alphacoronaviruses and Betacoronaviruses have a growing potential for causing global epidemics. The fact the the current pandemic is caused by Betacoronavirus might be attributed to the more effective airborne way of transmission in comparison to alimentary way of transmission for Alphacoronaviruses. However, recent studies indicates the risks associated with emerging Alphacoronaviruses in swines with a potential to infect humans (Edwards et al., 2020). Epidemics in wild and domestic animals can provide a suitable reservoir and required supply of computational resources for obtaining necessary genomic changes for viral adaptation to new hosts including humans.

Since we usually do not monitor viral computational potential and available resources, it is just a matter of time for a global pandemic in humans to occur. To avoid it, we need to monitor reservoirs of viruses that came very close to humans including local epidemics of related viruses in humans (useful clues to overcome human defense) and viral sources of global animal pandemics especially those that affect lungs and immunity (necessary resources for powerful distributed computing). Also, we have to look for a bridge closing the gap between viruses that have clues to overcome human defenses and their access to large computational reservoirs to broadcast these findings.

Viruses always try to unlock defenses. Breaking human defenses is somewhat similar to breaking computer defenses. Both are a matter of time and available computational resources. Any code can potentially be broken with enough time and effort. Thus, we should closely monitor attempts of viruses to get access to powerful distributed computing, e.g., local epidemics and animal pandemics. Also, we must track the direction of the attacks and evaluate the intensity of these attacks. It gives us the opportunity to develop and prepare our defense strategies accordingly. It is true for biological viruses and computer viruses.

The proposed methods can be used for a large-scale analysis to predict possible

threats. In the future study, we plan to adapt these methods to monitoring of the pandemic and local epidemics.

7.6 Conclusion

Computational analysis of available genome data is a useful method to screen for important changes in viral genomes. The proposed innovative computational approach based on matrix methods and genomic dictionaries allow researchers to predict possible pandemics. Further study may help justify the prognostic value of these methods for identifying emerging viruses and track significant changes in the known viruses that might affect humans. The computational analysis of the observed similarities between Coronaviruses allows us to formulate three necessary conditions that need to be met to initiate a global threat to public health similar to the COVID-19 pandemic: (1) susceptible hosts to Coronaviruses from multiple genera; (2) the presence of sparks of local epidemics of Coronaviruses adapted to humans and (3) the current reservoir for accelerated viral evolution such as a viral pandemic in animals with transmission ways to humans or other animal species that can contact with humans. If these conditions are met the avalanche of the next pandemic is on its way.

Chapter 8

Conclusion and future work

This study indicates that dictionary-based methods are a powerful computational approach for analyzing unannotated genome data. We applied the developed approach to address a number of challenges in biology and medicine including (1) to explore host-parasite associations; (2) to predict organism's functional properties (e.g., pathogenisity in bacteria); (3) to investigate autoimmunity potential in prokaryotes, and (4) to capture important changes in viral genomes for predicting possible epidemics. The developed methods are scalable and effective for large-scale screening research. This allows researchers to explore and analyze large amounts of genome data that are constantly added to the databases. These methods shift the computational paradigm from a partial to holistic view on genomes. Further development of holistic methods will help better understand genome organization and interactions between genomes.

This study aims to broaden the biological research focus on considering mainly genes and other annotated entities to analyzing entire genomes. Importantly, current genome databases become a sufficient resource for preliminary screening search. Now it is possible to apply pure computational methods to accumulated genome data to obtain reliable conclusions about biological objects and their interactions. We use random simulation and statistical modeling to adjust the level of resolution (sensi-

tivity and specificity) of the developed computational methods to efficiently separate signals from noise in genome data. This genome-scale view allows us to frame a complete picture of potential interactions without being restricted to specific annotated entities such as known genes. Our preliminary results demonstrate that it is possible to effectively apply machine learning methods to these big data to predict functional behavior of interacting organisms. It also allows us to characterize interactions between organisms at a broader scale and can help better understand the mechanisms involved. All of the above provides an opportunity to look at genome data from a new perspective. The use of a holistic approach in computational research is important from a philosophical perspective for proper design of experiments and comprehensive interpretation of the results. The consequences of this paradigm shift can advance our understanding of genome organization and genome interactions. In turn, it can facilitate broader applications of computational methods for discerning structural and functional properties of interacting organisms in biology and medicine.

References

Abedon, S., Kuhl, S., Blasdel, B., and Kutter, E. (2011). Phage treatment of human infections. bacteriophage 1: 66–85.

Ahlgren, N. A., Ren, J., Lu, Y. Y., Fuhrman, J. A., and Sun, F. (2017). Alignment-free oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences. *Nucleic acids research*, 45(1):39–53.

Al-Ghalith, G. and Knights, D. (2017). Burst enables optimal exhaustive dna alignment for big data. *DOI: doi. org/10.5281/zenodo*, 806850.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.

Anderson, K. (January 24, 2020). nCoV-2019 codon usage and reservoir (not snakes v2). *Virological.org*, http://virological.org/t/339.

Arber, W. (1978). Restriction endonucleases. *Angewandte Chemie International Edition in English*, 17(2):73–79.

Arndt, D., Grant, J. R., Marcu, A., Sajed, T., Pon, A., Liang, Y., and Wishart, D. S. (2016). Phaster: a better, faster version of the phast phage search tool. *Nucleic acids research*, 44(W1):W16–W21.

Barrangou, R., Fremaux, C., Deveau, H., Richards, M., Boyaval, P., Moineau, S., Romero, D. A., and Horvath, P. (2007). Crispr provides acquired resistance against viruses in prokaryotes. *Science*, 315(5819):1709–1712.

Barrangou, R. and Van Der Oost, J. (2013). Crispr-cas systems. *RNA-Mediated Adaptative Immunity in Bacteria and Archaea. 1010079783642346576th ed. Heidelberg SVB, editor*.

Bartoszek, K., Majchrzak, M., Sakowski, S., Kubiak-Szeligowska, A. B., Kaj, I., and Parniewski, P. (2018). Predicting pathogenicity behavior in escherichia coli population through a state dependent model and trs profiling. *PLoS computational biology*, 14(1):e1005931.

Besser, R. E., Lett, S. M., Weber, J. T., Doyle, M. P., Barrett, T. J., Wells, J. G., and Griffin, P. M. (1993). An outbreak of diarrhea and hemolytic uremic syndrome from escherichia coli o157: H7 in fresh-pressed apple cider. *Jama*, 269(17):2217–2220.

Blaisdell, B. E. (1986). A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences*, 83(14):5155–5159.

Blanco, J., Blanco, M., Blanco, J. E., Mora, A., Alonso, M. P., Gonzalez, E. A., and Bernardez, M. I. (2001). Epidemiology of verocytotoxigenic escherichia coli (vtec) in ruminants. *Verocytotoxigenic Escherichia coli*, pages 113–148.

Bolduc, B., Jang, H. B., Doulcier, G., You, Z.-Q., Roux, S., and Sullivan, M. B. (2017). vcontact: an ivirus tool to classify double-stranded dna viruses that infect archaea and bacteria. *PeerJ*, 5:e3243.

Boni, M. F., Lemey, P., Jiang, X., Lam, T. T.-Y., Perry, B., Castoe, T., Rambaut, A.,

and Robertson, D. L. (2020). Evolutionary origins of the SARS-CoV-2 sarbecovirus lineage responsible for the COVID-19 pandemic. *Nature Microbiology*.

Breiman, L. (2018). randomforest: Breiman and cutler's random forests for classification and regression, version 4.6.

Breiman, L. and Cutler, A. (2007). Random forests-classification description. *Department of Statistics, Berkeley*, 2.

Briner, A. E. and Barrangou, R. (2014). Lactobacillus buchneri genotyping on the basis of clustered regularly interspaced short palindromic repeat (crispr) locus diversity. *Applied and environmental microbiology*, 80(3):994–1001.

Brüssow, H., Canchaya, C., and Hardt, W.-D. (2004). Phages and the evolution of bacterial pathogens: from genomic rearrangements to lysogenic conversion. *Microbiology and molecular biology reviews*, 68(3):560–602.

Carroll, D., Daszak, P., Wolfe, N. D., Gao, G. F., Morel, C. M., Morzaria, S., Pablos-Méndez, A., Tomori, O., and Mazet, J. A. (2018). The global virome project. *Science*, 359(6378):872–874.

Casjens, S. (2003). Prophages and bacterial genomics: what have we learned so far? *Molecular microbiology*, 49(2):277–300.

Castellini, A., Franco, G., and Manca, V. (2012). A dictionary based informational genome analysis. *BMC genomics*, 13(1):485.

Chattaway, M. A., Schaefer, U., Tewolde, R., Dallman, T. J., and Jenkins, C. (2017). Identification of escherichia coli and shigella species from whole-genome sequences. *Journal of clinical microbiology*, 55(2):616–623.

Chen, Y., Ye, W., Zhang, Y., and Xu, Y. (2015). High speed blastn: an accelerated megablast search tool. *Nucleic acids research*, 43(16):7762–7768.

Compeau, P. E., Pevzner, P. A., and Tesler, G. (2011). How to apply de bruijn graphs to genome assembly. *Nature biotechnology*, 29(11):987–991.

Cowley, L. A., Beckett, S. J., Chase-Topping, M., Perry, N., Dallman, T. J., Gally, D. L., and Jenkins, C. (2015). Analysis of whole genome sequencing for the escherichia coli o157: H7 typing phages. *BMC genomics*, 16(1):271.

de Filippo, C., Meyer, M., and Prüfer, K. (2018). Quantifying and reducing spurious alignments for the analysis of ultra-short ancient dna sequences. *BMC biology*, 16(1):1–11.

Delcher, A. L., Kasif, S., Fleischmann, R. D., Peterson, J., White, O., and Salzberg, S. L. (1999). Alignment of whole genomes. *Nucleic acids research*, 27(11):2369–2376.

Dunne, K. A., Chaudhuri, R. R., Rossiter, A. E., Beriotto, I., Browning, D. F., Squire, D., Cunningham, A. F., Cole, J. A., Loman, N., and Henderson, I. R. (2017). Sequencing a piece of history: complete genome sequence of the original escherichia coli strain. *Microbial genomics*, 3(3).

Dutta, C. and Pan, A. (2002). Horizontal gene transfer and bacterial diversity. *Journal of biosciences*, 27(1):27–33.

Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics (Oxford, England)*, 14(9):755–763.

Edwards, C. E., Yount, B. L., Graham, R. L., Leist, S. R., Hou, Y. J., Dinnon, K. H., Sims, A. C., Swanstrom, J., Gully, K., Scobey, T. D., et al. (2020). Swine acute

diarrhea syndrome coronavirus replication in primary human cells reveals potential susceptibility to infection. *Proceedings of the National Academy of Sciences*, 117(43):26915–26925.

Edwards, R. A., McNair, K., Faust, K., Raes, J., and Dutilh, B. E. (2016). Computational approaches to predict bacteriophage–host relationships. *FEMS microbiology reviews*, 40(2):258–272.

Escalera-Zamudio, M. and Greenwood, A. D. (2016). On the classification and evolution of endogenous retrovirus: human endogenous retroviruses may not be 'human'after all. *Apmis*, 124(1-2):44–51.

Escherich, T. (1988). The intestinal bacteria of the neonate and breast-fed infant. *Clinical Infectious Diseases*, 10(6):1220–1225.

Fouts, D. E. (2006). Phage_finder: automated identification and classification of prophage regions in complete bacterial genome sequences. *Nucleic acids research*, 34(20):5839–5851.

Gaj, T. et al. (2020). Next-generation crispr technologies and their applications in gene and cell therapy. *Trends in Biotechnology*.

Galiez, C., Siebert, M., Enault, F., Vincent, J., and Söding, J. (2017). Wish: who is the host? predicting prokaryotic hosts from metagenomic phage contigs. *Bioinformatics*, 33(19):3113–3114.

Garcia, E., Chain, P., Elliott, J. M., Bobrov, A. G., Motin, V. L., Kirillina, O., Lao, V., Calendar, R., and Filippov, A. A. (2008). Molecular characterization of l-413c, a p2-related plague diagnostic bacteriophage. *Virology*, 372(1):85–96.

George, A. and Liu, J. W. (1989). The evolution of the minimum degree ordering algorithm. *Siam review*, 31(1):1–19.

Gibbs, A. J. and McIntyre, G. A. (1970). The diagram, a method for comparing sequences: Its use with amino acid and nucleotide sequences. *European journal of biochemistry*, 16(1):1–11.

Goeddel, D. V., Kleid, D. G., Bolivar, F., Heyneker, H. L., Yansura, D. G., Crea, R., Hirose, T., Kraszewski, A., Itakura, K., and Riggs, A. D. (1979). Expression in escherichia coli of chemically synthesized genes for human insulin. *Proceedings of the National Academy of Sciences*, 76(1):106–110.

Grad, Y. H., Lipsitch, M., Feldgarden, M., Arachchi, H. M., Cerqueira, G. C., FitzGerald, M., Godfrey, P., Haas, B. J., Murphy, C. I., Russ, C., et al. (2012). Genomic epidemiology of the escherichia coli o104: H4 outbreaks in europe, 2011. *Proceedings of the national academy of sciences*, 109(8):3065–3070.

Grau, J., Keilwagen, J., Gohr, A., Haldemann, B., Posch, S., and Grosse, I. (2012). Jstacs: a java framework for statistical analysis and classification of biological sequences. *The Journal of Machine Learning Research*, 13(1):1967–1971.

Griffiths, D. J. (2001). Endogenous retroviruses in the human genome sequence. *Genome biology*, 2(6):1–5.

Grissa, I., Vergnaud, G., and Pourcel, C. (2007). The crisprdb database and tools to display crisprs and to generate dictionaries of spacers and repeats. *BMC bioinformatics*, 8(1):172.

Guttinger, S. and Love, A. C. (2020). modENCODE and the elaboration of functional genomic methodology. In: C. Donohue and A.C. Love (eds.). *Perspectives on the Human Genome Project and Genomics. Minnesota Studies in Philosophy of Science*, Minneapolis: University of Minnesota Press.

Hand, D. J. and Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2):171–186.

Hatcher, E. L., Zhdanov, S. A., Bao, Y., Blinkova, O., Nawrocki, E. P., Ostapchuck, Y., Schäffer, A. A., and Brister, J. R. (2017). Virus variation resource–improved response to emergent viral outbreaks. *Nucleic acids research*, 45(D1):D482–D490.

Hirschberg, D. S. (1977). Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM)*, 24(4):664–675.

Howard-Varona, C., Hargreaves, K. R., Abedon, S. T., and Sullivan, M. B. (2017). Lysogeny in nature: mechanisms, impact and ecology of temperate phages. *The ISME journal*, 11(7):1511–1520.

Huttenhower, C., Gevers, D., Knight, R., Abubucker, S., Badger, J. H., Chinwalla, A. T., Creasy, H. H., Earl, A. M., FitzGerald, M. G., Fulton, R. S., et al. (2012). Structure, function and diversity of the healthy human microbiome. *nature*, 486(7402):207.

Iranzo, J., Krupovic, M., and Koonin, E. V. (2017). A network perspective on the virus world. *Communicative & integrative biology*, 10(2):e00978–16.

Ishino, Y., Krupovic, M., and Forterre, P. (2018). History of crispr-cas from encounter with a mysterious repeated sequence to genome editing technology. *Journal of bacteriology*, 200(7).

Ji, W., Wang, W., Zhao, X., Zai, J., and Li, X. (2020). Cross-species transmission of the newly identified coronavirus 2019-nCoV. *Journal of medical virology*, 92(4):433–440.

Kalscheuer, R., Stölting, T., and Steinbüchel, A. (2006). Microdiesel: Escherichia coli engineered for fuel production. *Microbiology*, 152(9):2529–2536.

Kaper, J. B., Nataro, J. P., and Mobley, H. L. (2004). Pathogenic escherichia coli. *Nature reviews microbiology*, 2(2):123–140.

Karlin, S., Mrazek, J., and Campbell, A. M. (1997). Compositional biases of bacterial genomes and evolutionary implications. *Journal of bacteriology*, 179(12):3899–3913.

Karp, R. (1987). Efficient randomized pattern-matching algorithms, the ibm journal of research and development. *http://www. research. ibm. com/journal/rd/312/ibmrd3102P. pdf*, 31.

Kim, J. H., Kalitsis, P., Pertile, M. D., Magliano, D., Wong, L., Choo, A., and Hudson, D. F. (2012). Nucleic acids: Hybridisation. *eLS*.

Kokot, M., Długosz, M., and Deorowicz, S. (2017). Kmc 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761.

Koonin, E. and Galperin, M. Y. (2002). *Sequence—evolution—function: computational approaches in comparative genomics*. Springer Science & Business Media.

Kuhn, M. (2018). Classification and regression training.

Kurtz, S., Narechania, A., Stein, J. C., and Ware, D. (2008). A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes. *BMC genomics*, 9(1):1–18.

Kutter, E. (2009). Phage host range and efficiency of plating. In *Bacteriophages*, pages 141–149. Springer.

Lam, T. T.-Y., Jia, N., Zhang, Y.-W., Shum, M. H.-H., Jiang, J.-F., Zhu, H.-C., Tong, Y.-G., Shi, Y.-X., Ni, X.-B., Liao, Y.-S., et al. (2020). Identifying SARS-CoV-2-related coronaviruses in Malayan pangolins. *Nature*, pages 1–4.

Larimore, S. and Davis, T. (2020). Symmetric approximate minimum degree permutation. *MathWorks Documentation*, MATLAB R2020b, https://www.mathworks.com/help/matlab/ref/symamd.html.

Lenskaia, T. and Boley, D. (2018). High-throughput phage screening to predict pathogenicity of *E.coli* strains. *ICML&IJCAI 2018*, Workshop on Computational Biology, July 14, Stockholm, Sweden.

Lenskaia, T. and Boley, D. (2019). Exploring mechanisms of genomic exchange between virulent phages and microbial hosts. *American Society for Virology (ASV) Annual Meeting*, June 20-24, University Minnesota, Twin Cities, MN.

Lenskaia, T. and Boley, D. (2020a). Prokaryote autoimmunity in the context of self-targeting by CRISPR-Cas systems. *Journal of Bioinformatics and Computational Biology*, 18(5).

Lenskaia, T. and Boley, D. (2020b). Scalable computational methods for predicting and preventing viral epidemics. *The Institute for Molecular Virology (IMV) Special Event "Minnesota's Response to COVID-19"*, 13 May, University of Minnesota, Twin Cities, MN.

Leplae, R., Hebrant, A., Wodak, S. J., and Toussaint, A. (2004). Aclame: a classification of mobile genetic elements. *Nucleic acids research*, 32(suppl_1):D45–D49.

Leslie, C., Eskin, E., and Noble, W. S. (2001). The spectrum kernel: A string kernel for svm protein classification. In *Biocomputing 2002*, pages 564–575. World Scientific.

Loenen, W. A., Dryden, D. T., Raleigh, E. A., Wilson, G. G., and Murray, N. E. (2014). Highlights of the dna cutters: a short history of the restriction enzymes. *Nucleic acids research*, 42(1):3–19.

Long, M. (2000). A new function evolved from gene fusion. *Genome research*, 10(11):1655–1657.

Manekar, S. C. and Sathe, S. R. (2018). A benchmark study of k-mer counting methods for high-throughput sequencing. *GigaScience*, 7(12):giy125.

Manrique, P., Bolduc, B., Walk, S. T., van der Oost, J., de Vos, W. M., and Young, M. J. (2016). Healthy human gut phageome. *Proceedings of the National Academy of Sciences*, 113(37):10400–10405.

Marçais, G. and Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770.

Marçais, G., Pellow, D., Bork, D., Orenstein, Y., Shamir, R., and Kingsford, C. (2017). Improving the performance of minimizers and winnowing schemes. *Bioinformatics*, 33(14):i110–i117.

Mayr, E. (1982). *The growth of biological thought: Diversity, evolution, and inheritance*. Harvard University Press.

McGinn, J. and Marraffini, L. A. (2019). Molecular mechanisms of crispr–cas spacer acquisition. *Nature Reviews Microbiology*, 17(1):7–12.

Melsted, P. and Pritchard, J. K. (2011). Efficient counting of k-mers in dna sequences using a bloom filter. *BMC bioinformatics*, 12(1):333.

Mokili, J. L., Rohwer, F., and Dutilh, B. E. (2012). Metagenomics and future perspectives in virus discovery. *Current opinion in virology*, 2(1):63–77.

Morgenstern, B. (1999). Dialign 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics (Oxford, England)*, 15(3):211–218.

Narlikar, L., Mehta, N., Galande, S., and Arjunwadkar, M. (2013). One size does not fit all: On how markov model order dictates performance of genomic sequence analyses. *Nucleic acids research*, 41(3):1416–1424.

Nasko, D. J., Ferrell, B. D., Moore, R. M., Bhavsar, J. D., Polson, S. W., and Wommack, K. E. (2019). Crispr spacers indicate preferential matching of specific virioplankton genes. *MBio*, 10(2).

Nuñez, J. K., Harrington, L. B., Kranzusch, P. J., Engelman, A. N., and Doudna, J. A. (2015). Foreign dna capture during crispr–cas adaptive immunity. *Nature*, 527(7579):535–538.

Ohno, S. (1972). So much'junk'dna in our genome. In *Evolution of Genetic Systems, Brookhaven Symp. Biol.*, pages 366–370.

Paez-Espino, D., Chen, I.-M. A., Palaniappan, K., Ratner, A., Chu, K., Szeto, E., Pillay, M., Huang, J., Markowitz, V. M., Nielsen, T., et al. (2016). Img/vr: a database of cultured and uncultured dna viruses and retroviruses. *Nucleic acids research*, page gkw1030.

Pantůček, R., Rosypalová, A., Doškař, J., Kailerová, J., Růžičková, V., Borecká, P., Snopková, Š., Horváth, R., GoÈtz, F., and Rosypal, S. (1998). The polyvalent staphylococcal phage φ812: its host-range mutants and related phages. *Virology*, 246(2):241–252.

Pearson, W. R. (1990). [5] rapid and sensitive sequence comparison with fastp and fasta. *Methods in Enzymology*, 183:63–98.

Penadés, J. R., Chen, J., Quiles-Puchalt, N., Carpena, N., and Novick, R. P. (2015). Bacteriophage-mediated spread of bacterial virulence genes. *Current opinion in microbiology*, 23:171–178.

Pevzner, P. A., Tang, H., and Waterman, M. S. (2001). An eulerian path approach to dna fragment assembly. *Proceedings of the national academy of sciences*, 98(17):9748–9753.

Pickar-Oliver, A. and Gersbach, C. A. (2019). The next generation of crispr–cas technologies and applications. *Nature reviews Molecular cell biology*, 20(8):490–507.

Pósfai, G., Plunkett, G., Fehér, T., Frisch, D., Keil, G. M., Umenhoffer, K., Kolisnychenko, V., Stahl, B., Sharma, S. S., De Arruda, M., et al. (2006). Emergent properties of reduced-genome escherichia coli. *science*, 312(5776):1044–1046.

Pourcel, C., Touchon, M., Villeriot, N., Vernadet, J.-P., Couvin, D., Toffano-Nioche, C., and Vergnaud, G. (2020). CRISPRCasdb a successor of CRISPRdb containing CRISPR arrays and cas genes from complete genome sequences, and tools to download and query lists of repeats and spacers. *Nucleic acids research*, 48(D1):D535–D544.

Prada, C. F. and Boore, J. L. (2019). Gene annotation errors are common in the mammalian mitochondrial genomes database. *BMC genomics*, 20(1):73.

Pride, D. T., Wassenaar, T. M., Ghose, C., and Blaser, M. J. (2006). Evidence of host-virus co-evolution in tetranucleotide usage patterns of bacteriophages and eukaryotic viruses. *BMC genomics*, 7(1):8.

Qi, J., Luo, H., and Hao, B. (2004). Cvtree: a phylogenetic tree reconstruction tool based on whole genomes. *Nucleic acids research*, 32(suppl_2):W45–W47.

Raetz, C. (1996). *Escherichia coli* and *Salmonella*: cellular and molecular biology. *Escherichia coli and Salmonella: Cellular and Molecular Biology*, 1:1035–1063.

Rangel, J. M., Sparling, P. H., Crowe, C., Griffin, P. M., and Swerdlow, D. L. (2005). Epidemiology of escherichia coli o157: H7 outbreaks, united states, 1982–2002. *Emerging infectious diseases*, 11(4):603.

Rauch, B. J., Silvis, M. R., Hultquist, J. F., Waters, C. S., McGregor, M. J., Krogan, N. J., and Bondy-Denomy, J. (2017). Inhibition of crispr-cas9 with bacteriophage proteins. *Cell*, 168(1-2):150–158.

Reinert, G., Chew, D., Sun, F., and Waterman, M. S. (2009). Alignment-free sequence comparison (i): statistics and power. *Journal of Computational Biology*, 16(12):1615–1634.

Ren, J., Ahlgren, N. A., Lu, Y. Y., Fuhrman, J. A., and Sun, F. (2017). Virfinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, 5(1):69.

Ren, J., Bai, X., Lu, Y. Y., Tang, K., Wang, Y., Reinert, G., and Sun, F. (2018). Alignment-free sequence analysis and applications. *Annual Review of Biomedical Data Science*, 1:93–114.

Robertson, D. (January 24, 2020). nCoV's relationship to bat coronaviruses & recombination signals (no snakes). *Virological.org*, https://virological.org/t/ncov-2019-codon-usage-and-reservoir-not-snakes-v2/339.

Roux, S., Enault, F., Hurwitz, B. L., and Sullivan, M. B. (2015). Virsorter: mining viral signal from microbial genomic data. *PeerJ*, 3:e985.

Sass, P., Berscheid, A., Jansen, A., Oedenkoven, M., Szekat, C., Strittmatter, A., Gottschalk, G., and Bierbaum, G. (2012). Genome sequence of staphylococcus aureus vc40, a vancomycin-and daptomycin-resistant strain, to study the genetics of development of resistance to currently applied last-resort antibiotics.

Scallan, E., Hoekstra, R. M., Angulo, F. J., Tauxe, R. V., Widdowson, M.-A., Roy, S. L., Jones, J. L., and Griffin, P. M. (2011). Foodborne illness acquired in the united states—major pathogens. *Emerging infectious diseases*, 17(1):7.

Schuler, G. D., Altschul, S. F., and Lipman, D. J. (1991). A workbench for multiple alignment construction and analysis. *Proteins: Structure, Function, and Bioinformatics*, 9(3):180–190.

Sievers, A., Bosiek, K., Bisch, M., Dreessen, C., Riedel, J., Froß, P., Hausmann, M., and Hildenbrand, G. (2017). K-mer content, correlation, and position analysis of genome dna sequences for the identification of function and evolutionary features. *Genes*, 8(4):122.

Snyder, M. P., Gingeras, T. R., Moore, J. E., Weng, Z., Gerstein, M. B., Ren, B., Hardison, R. C., Stamatoyannopoulos, J. A., Graveley, B. R., Feingold, E. A., et al. (2020). Perspectives on encode. *Nature*, 583(7818):693–698.

Stein, L. (2001). Genome annotation: from sequence to biology. *Nature reviews genetics*, 2(7):493–503.

Stern, A., Keren, L., Wurtzel, O., Amitai, G., and Sorek, R. (2010). Self-targeting by crispr: gene regulation or autoimmunity? *Trends in genetics*, 26(8):335–340.

Stern, A., Mick, E., Tirosh, I., Sagy, O., and Sorek, R. (2012). Crispr targeting reveals a reservoir of common phages associated with the human gut microbiome. *Genome research*, 22(10):1985–1994.

Swanson, M. M., Reavy, B., Makarova, K. S., Cock, P. J., Hopkins, D. W., Torrance, L., Koonin, E. V., and Taliansky, M. (2012). Novel bacteriophages containing a genome of another bacteriophage within their genomes. *PloS one*, 7(7):e40683.

Taft, R. J., Pheasant, M., and Mattick, J. S. (2007). The relationship between non-protein-coding dna and eukaryotic complexity. *Bioessays*, 29(3):288–299.

Teeling, H., Waldmann, J., Lombardot, T., Bauer, M., and Glöckner, F. O. (2004). Tetra: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in dna sequences. *BMC bioinformatics*, 5(1):163.

Touchon, M., Bernheim, A., and Rocha, E. P. (2016). Genetic and life-history traits associated with the distribution of prophages in bacteria. *The ISME journal*, 10(11):2744–2754.

Tsangaras, K., Siracusa, M. C., Nikolaidis, N., Ishida, Y., Cui, P., Vielgrader, H., Helgen, K. M., Roca, A. L., and Greenwood, A. D. (2014). Hybridization capture reveals evolution and conservation across the entire koala retrovirus genome. *PloS one*, 9(4):e95633.

Ventola, C. L. (2015). The antibiotic resistance crisis: part 1: causes and threats. *Pharmacy and therapeutics*, 40(4):277.

Villarroel, J., Kleinheinz, K. A., Jurtz, V. I., Zschach, H., Lund, O., Nielsen, M., and Larsen, M. V. (2016). Hostphinder: a phage host prediction tool. *Viruses*, 8(5):116.

Vinga, S. and Almeida, J. (2003). Alignment-free sequence comparison—a review. *Bioinformatics*, 19(4):513–523.

Wang, H. (2007). All common subsequences. In *IJCAI*.

Wassenaar, T. M. and Gunzer, F. (2015). The prediction of virulence based on presence of virulence genes in e. coli may not always be accurate. *Gut pathogens*, 7(1):1–3.

Watters, K. E., Fellmann, C., Bai, H. B., Ren, S. M., and Doudna, J. A. (2018). Systematic discovery of natural crispr-cas12a inhibitors. *Science*, 362(6411):236–239.

WHO (2018). A global strategy to eliminate yellow fever epidemics (EYE) 2017–2026.

Williamson, A. and Leiros, H.-K. S. (2020). Structural insight into dna joining: from conserved mechanisms to diverse scaffolds. *Nucleic acids research*, 48(15):8225–8242.

Wu, F., Zhao, S., Yu, B., Chen, Y.-M., Wang, W., Song, Z.-G., Hu, Y., Tao, Z.-W., Tian, J.-H., Pei, Y.-Y., et al. (2020). A new coronavirus associated with human respiratory disease in China. *Nature*, 579(7798):265–269.

Xiao, K., Zhai, J., Feng, Y., Zhou, N., Zhang, X., Zou, J.-J., Li, N., Guo, Y., Li, X., Shen, X., et al. (2020). Isolation of SARS-CoV-2-related coronavirus from Malayan pangolins. *Nature*, pages 1–4.

Zhang, M., Yang, L., Ren, J., Ahlgren, N. A., Fuhrman, J. A., and Sun, F. (2017). Prediction of virus-host infectious association by supervised learning methods. *BMC bioinformatics*, 18(3):60.

Zhou, P., Yang, X.-L., Wang, X.-G., Hu, B., Zhang, L., Zhang, W., Si, H.-R., Zhu, Y., Li, B., Huang, C.-L., et al. (2020). A pneumonia outbreak associated with a new coronavirus of probable bat origin. *Nature*, 579(7798):270–273.